INFORMATION SYSTEMS LABORATORY

STANFORD ELECTRONICS LABORATORIES
DEPARTMENT OF ELECTRICAL ENGINEERING
STANFORD UNIVERSITY · STANFORD, CA 94305

SEL-79-010

INVESTIGATION OF NEW ALGORITHMS FOR LOCATING AND IDENTIFYING SPATIALLY DISTRIBUTED SOURCES AND RECEIVERS

TECHNICAL REPORT NO. M355-1

Annual Technical Summary Report to the

Defense Advanced Research Project Agency
Contract Number: DARPA MDA 903-78-C-0179

15 December 1977 - 15 December 1978

Issued: 31 March 1979

INVESTIGATION OF NEW ALGORITHMS FOR LOCATING AND IDENTIFYING SPATIALLY DISTRIBUTED SOURCES AND RECEIVERS

DRAFT

Annual Technical Summary Report to the

Defense Advanced Research Project Agency

Contract Number: DARPA MDA 903-78-C-0179

15 December 1977 - 15 December 1978

Issued: 31 March 1979

Project Personnel

Principal Investigators

Prof. M. Morf,

Prof. T. Kailath

Contributors:

Dr. B. Friedlander (Systems Control Inc.)

J. Newkirk, Research Associate

D. T. Lee, Pre-Doctoral Research Affiliate

E. Verriest, Research Assistant

J. M. Delosme, Research Assistant

Table of Contents

1. Introduction and Summary

This report describes the research activities in the Information Systems Laboratory (ISL) at Stanford Universty on Distributed Sensor Networks (DSN). Our effort is part of a DARPA-sponsored program with participation by several contractors. The objectives are to develop new and innovative signal processing and computer network techniques with applications to systems employing multiple sensors for target surveillance and tracking. Such systems would be composed of sensors, processors, and data bases distributed throughout an area, interconnected by a suitable communication system. The system will serve a user-community that is also distributed and connected to the same communication system.

A basic premise of our approach is that most of the currently available signal processing algorithms are not well suited to the DSN problem because of their highly centralized structure. The straightforward application of standard techniques in the context of distributed networks leads to ad hoc, suboptimal designs. We feel that it is essential to have a more careful look at the basic requirements for developing and implementing distributed algorithms. Our preliminary conclusions were that improved system performance and a more natural system structure will result if innovative signal processing modules will be used; new analytical and computational techniques have to be developed and used in the DSN design, rather than trying to adapt this problem to standard solutions. Accordingly, the major emphasis of our research effort to date has been the development of novel signal processing algorithms which are especially suited to the DSN.

The impact of these optimal distributed algorithms on the overall network design was also addressed. This effort is currently in the developmental stage, but some very promising preliminary results have already been obtained.

DSN System Structure

The Distributed Sensor Network contains various classes of sensors and is required to provide different categories of user information. Therefore, the DSN system must have a collection of signal processing modules which can be used in different combinations according to the sensors adopted and the type of information requested.

Another feature of the DSN is the wide range of data rates appearing in different system components: from the wideband "raw" sensor data to the relatively low bandwidth summary information required by the user. A related aspect is the communication bandwidth requirements between different system components. A local set of sensors may have to exchange high-rate data to perform the necessary signal processing (e.g. cross correlation); sensors from separate regions may only need to exchange low-rate target parameters (e.g. bearing and range estimates) to generate location/velocity estimates. Finally, the communication between the user and the network may be at even lower rates.

This type of basic consideration has led us to a hierarchical system structure, where different levels have different signal processing modules.

At the top are the user entry points and inter-user communication. The interface between the user and the network also includes more complex functions such as combining and synthesizing data sets provided from different regions.

At the next level are the regional processors which can assemble the answers to user questions required on a local level: e.g. combine bearing measurements from several data sources to give target location estimates.

Parameter-list processing is the next lower level in the system. Here, some initial target parameters are processed to provide higher quality information. For example: time-difference-of-arrival (TDOA) information is translated into range/bearing estimates. This level performs relatively complex computations, so as to provide the regional processor with easily usable data. Different parameter-list processors will communicate at moderate data rates.

It should be noted that the proper choice of parameters to be used by this processor (both input and output) has a major impact on system performance and communication/computation cost. An important part of our work was devoted to developing and identifying appropriate parametrizations for the target location problem.

At the lowest level are the front-end processors which operate directly on the "raw" sensor data. These processors provide relatively low quality target data parameters which are then processed by the parameter-list processor. Local sensor sites may need to communicate at high bandwidth to compute these parameter estimates. The sensor together with its front-end processors can be thought of as a single "smart sensor" unit.

Signal Processing Modules

The ISL project group has been responsible for a number of novel theoretical developments in the area of algorithms for signal processing, estimation - identification and control. We have applied these approaches to the development of several signal processing modules, which will provide the components for the DSN system described earlier. These include:

- Real-Time ARMA and Delay-Differential System Modeling.

  This is a new approach which provides a unifying framework to multi-target bearing and spectral signature estimation.

- The Linear Equation Approach to the Target Location Problem.

  Our approach provides efficient algorithms for computing target location from time-of-arrival differences, in a distributed fashion.

- Distributed Kalman Filtering Algorithms.

  Some ideas from scattering theory were used to derive new distributed versions of the Kalman filter applicable to the target tracking problem.

- Image Reconstruction Techniques.

  Algorithms which were developed to reconstruct images from projections (e.g. for computer assisted tomography, in medical applications) are used to estimate target location from lists of partial target information (e.g. bearing only or range only).

- Event Detection.

  An innovative approach to handling inordinate amounts of sensor data, this class of algorithms identifies significant "events" in the data (e.g. appearance of a new target, or change of course of a target under track), and limits the processing effort to the neighborhood of these events.

- Two- and Three-Dimensional Signal Processing.

  This is a mathematical framework which treats the target location problem as a multi-dimensional imaging problem.

The rest of this report contains:

(i) A more detailed description of our overall approach to the distributed sensor network organization, section 2:

Distributed Sensor Network Design.

(ii) An overview of the signal processing modules which have been developed or are currently under development. These modules are the building blocks of our DSN system design, section 3:

Signal Processing Modules for DSN.

(iii) A collection of three papers presenting the details of some of our processing modules:

A System Identification Approach to Locating Spatially Distributed Targets.

Source Location from Time Difference of Arrival: A Linear Equation Approach.

Distributed Algorithms for Estimation and Detection.

2. Distributed Sensor Network Design

The detailed design of a sensor network depends intimately upon the choice of sensors, the characteristics of the input signals, and the specifications of the output requirements. Thus, only the outline of a system will be proposed. Several classes of sensors will be considered, and alternative algorithms will be introduced, to be selected later depending upon the characteristics of the input signal.

An important caveat is that this is our current best guess at an organization for such a network, based upon our recent work in multichannel and distributed algorithms. Some aspects of this design are well understood, others are very poorly understood. For example, performance bounds of individual algorithms can be readily obtained; the interactions and overall performance of systems of algorithms, however, are much harder to predict. In addition, the proposed partitioning into modules may be obviated by future work.

Figure 1 presents a preliminary system organization. User output consists of target location and, perforce, target detection. [Other target information, such as reflection coefficients, acoustic spectra, etc., will also be available as secondary results.] Input derives from three classes of sensors: omni-directional, single-dimensional, and multi-dimensional. Omni-directional sensors are rich in information content but provide only indirect location data. The single-dimensional sensors provide, for example, range-only or bearing-only information. This information must then be integrated into the data from the third class of sensors, which provides location information (e.g., range and bearing) directly. The integration process should take into account that measurements have varying error probability distributions or, in the worst case, are erroneous.

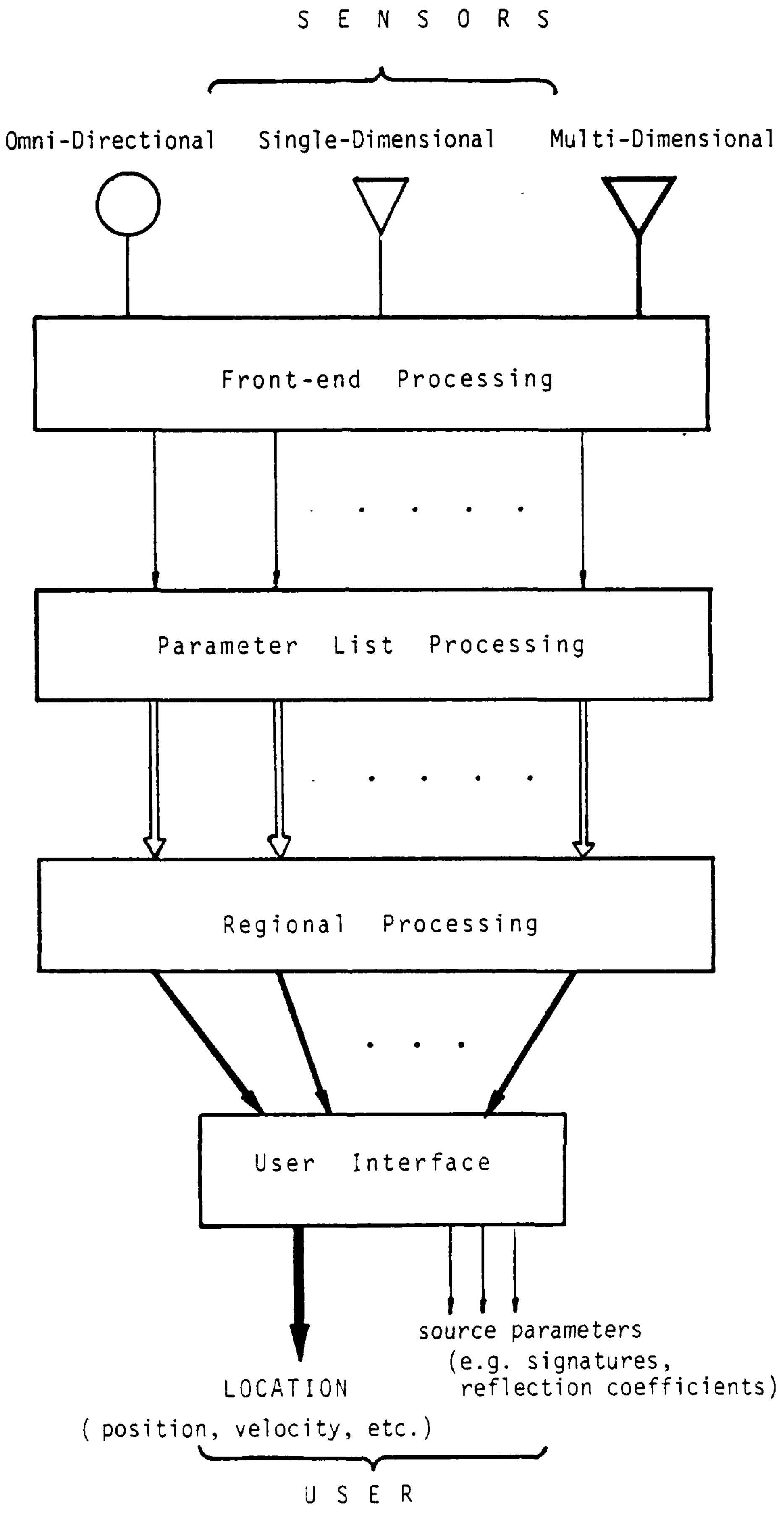


Figure 1. System Block Diagram

The processing can be partitioned into four stages: front-end sensor processing, parameter-list processing, regional processing and the user interface. Front-end processing transforms raw sensor data into target paramters suitable for refinement at later stages. For example, the omni-directional sensor processors could provide, as output, range, bearing, or time-of-arrival (TOA) information. This step typically requires tight, high-bandwidth coupling between sensor processors if, for example, cross-correlation is required.

The parameter-list stage accepts lists of target-parameters as input and produces, as output, components of the location. This processing step typically requires information from a local neighborhood of sensors; a good example would be the synthesis of target location from time-of-arrival input.

At the third stage, information from regional groups of sensors is combined. The regions may be large, and the information is typically imperfect and redundant. The final stage then integrates all information into an output desired by the user community.

Front-End Processing

Much of the front-end processing is best described in the context of developing an algorithm, such as the Extended Kalman Filter, which cuts across many stages of processing. As shown in figure 2, however, the front-end processing required for omni-directional sensors can be considered as an integral unit.

The frequency-wavenumber block -- which produces, as output *location information* -- and the long-baseline cross-correlation (without pre-whitening) -- which produces *time-difference-of-arrival* (TDOA) data -- were presented by Lincoln Laboratories in their initial strawman proposal (Lincoln [78]).

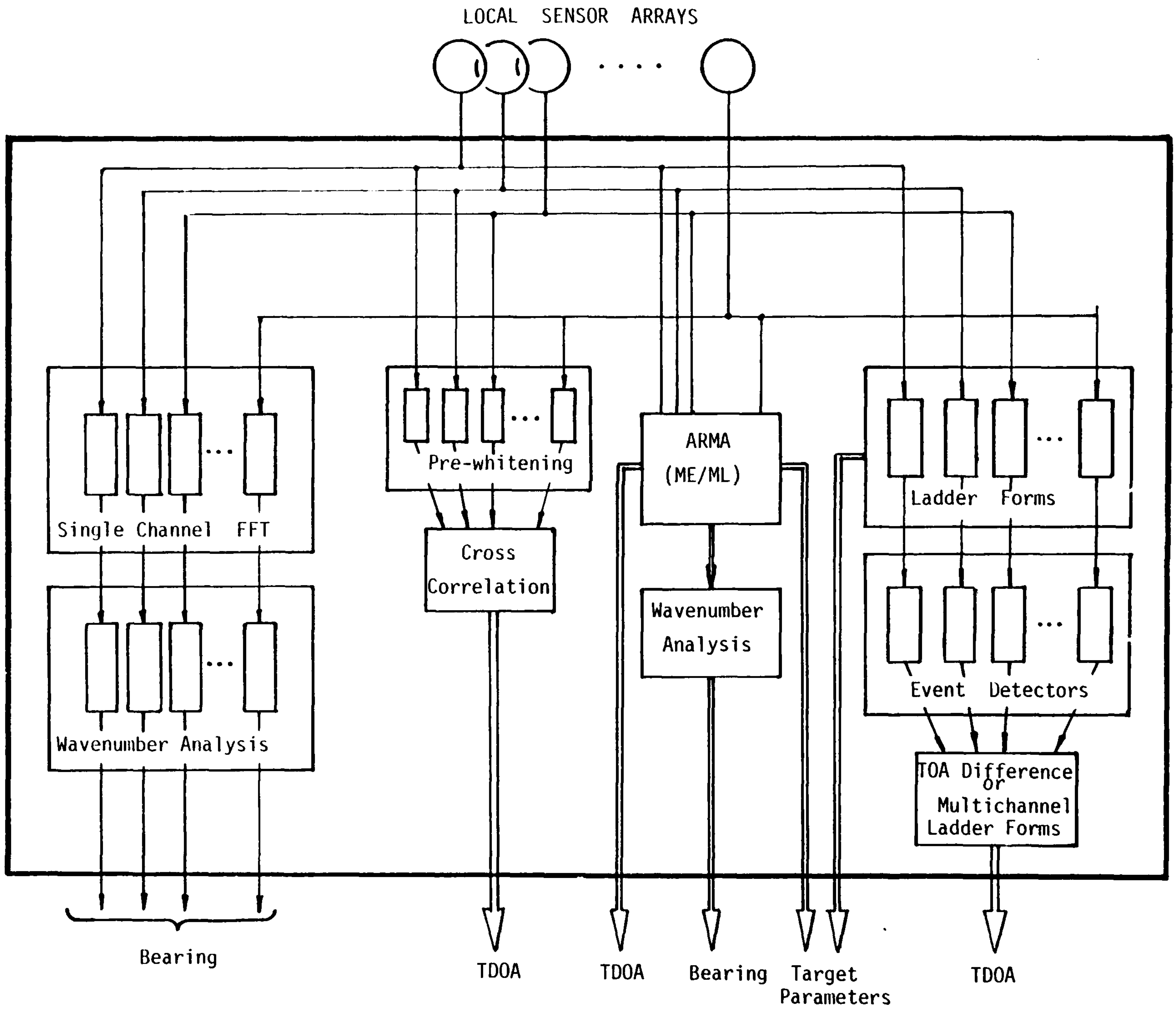


Figure 2. Front-end Signal Processing for Omni-Directional Sensors

The ARMA modeling block (see Section IV and Appendix B of [FM]) is a refinement and unification of these approaches. Using high-resolution, spectral estimation (maximum likelihood or maximum entropy techniques) would refine the data provided by the strawman algorithms. More precise bearing and TDOA data will be available, as well as a host of other auxiliary parameters.

The fourth block (see also figure 3), using ladder forms, begins with a data compression stage. Each sensor's data is processed independently to detect discernable changes in the data (events). compression is a two step procedure: first, the data passes through ladder-form filters which identify and locate changes, producing a log-likelihood function; second, an event detector scans this function, looking for significant events and thereby generating TOA data. If the events are precise enough this data can be combined trivially to get TDOA information; alternatively, the raw-data near the event can be exchanged among sensors and a multi-channel ladder-form filter can be used to generate a joint log-likelihood function, and TDOA information thereby.

[The existence of "events" or similar signal characteristics is largely an unexplored question at this moment. Preliminary analysis of various signal signal sources has shown that speech and certain other acoustic signals display "events" (in speech these could be either plosive sounds or voiced speech). The recent availability of our efficient recursive ladder-form algorithms enabled the "discovery process" of such events.]

These four algorithms, which convert omni-directional data into components of a location vector, can be compared along several dimensions: the expected characteristics of the input signal, the accuracy required in the output, the bandwidth required for inter-sensor communication, and the algorithm's amenability to distribution.

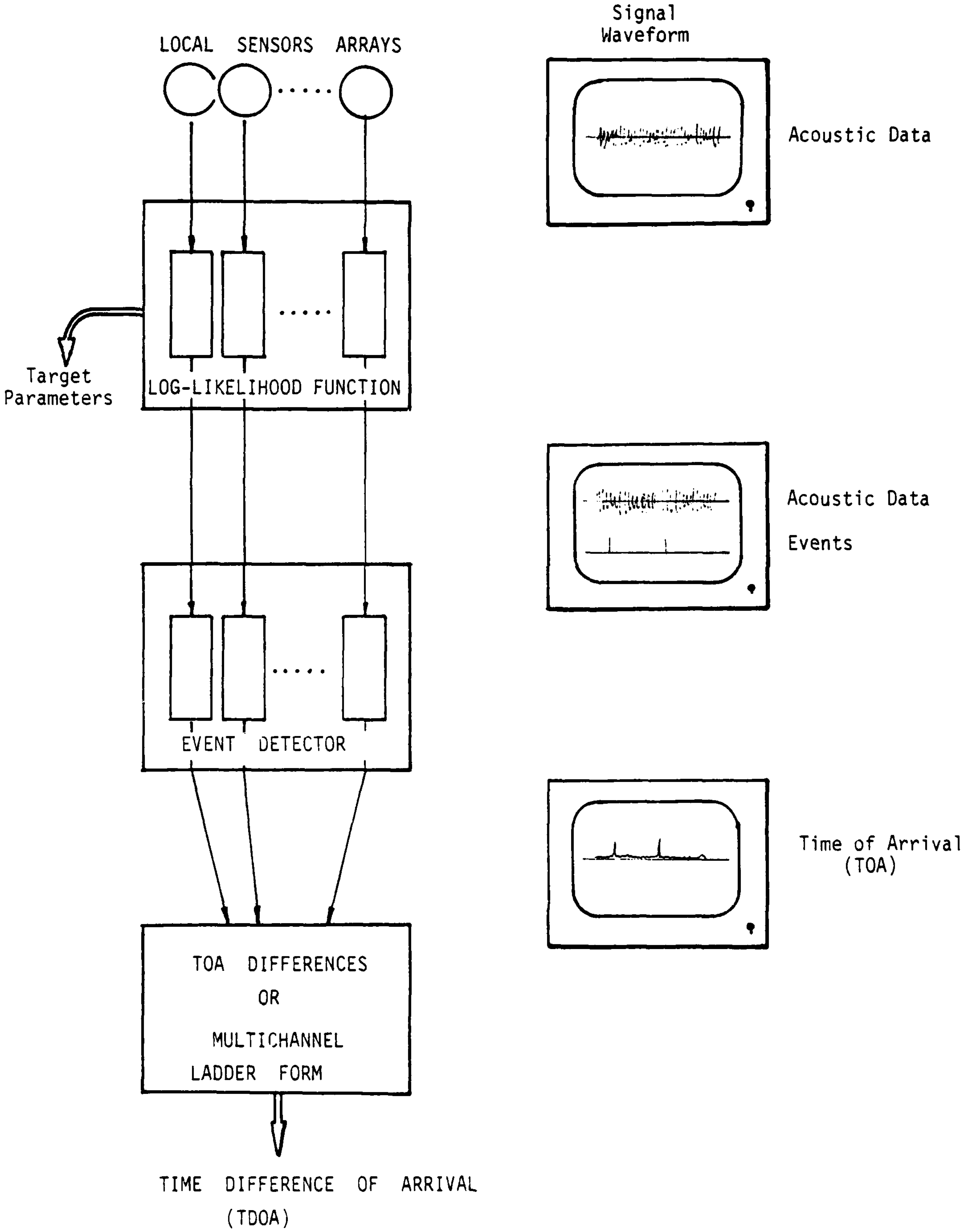


Figure 3. Front-end Signal Processing Using Event Detector

The frequency-wavenumber processing of local arrays of sensors, as proposed by Lincoln Laboratories, is a good example of an independently distributed algorithm that requires minimal communication between sensor sites. The FFT approach, however, has several drawbacks, since information is thrown away (cross-correlation data) and since the resolution is limited because of the effect of windowing and finite record length.

Long-baseline cross-correlation, in contrast, distributes very poorly, and requires very high inter-sensor communication bandwidth. Also, for the Lincoln Laboratory study, the algorithm produced poor results. Pre-whitening the data on input may improve the output this block, but long-baseline inteferometry processing will still require high-bandwidth inter-sensor exchange of the raw data.

The ARMA modeling is an intermediate solution, which replaces both the FFT and the cross-correlation processes; as would be expected, this approach also typically requires high-bandwith data exchange. However, these algorithms promise to yield superior results; assumptions concerning periodicity and the noise spectrum have not been made, and time-varying sources can be tracked. In addition, and in contrast to FFT and cross-correlation algorithms, ARMA modeling produces a minimal parameterization (parsimonious output), introducing well behaved data reduction at this early stage of analysis.

The communication drawback associated with ARMA modeling can be partially alleviated if we can use Matrix Fraction Description Models (MFD) in the ARMA block (see section 4). Then, the number of variables communicated is neither a function of the number of sensors nor of the amount of raw sensor data, but rather is a function of the needs of the user (i.e., of later stages of processing).

The fourth block distributes well and promises good communication characteristics (as can be seen from figure 3); the compression process implies only exchanging data about significant events, and possibly the raw data immediately surrounding those events. The first two stages are independently distributed, and require no inter-process communication. The output is raw sensor data, but the event detector only produces packets immediately surrounding statistically significant events. The final step, the determination of TDOA information, requires inter-process communication, but if the quality of events is very high, only the time of the events needs to be communicated; in the worst case, the packets of raw data would be exchanged, which would still have a comparatively low-bandwidth. Thus, if these algorithms are applicable to specific instantiations of the DSN concept, then we will have the high-resolution of an ARMA model with a very high degree of distribution and comparably low communication bandwidth requirements.

In summary, turning again to figure 2, we see that raw sensor data provides input to the front-end processors. As output, we receive parameter-list information, often redundantly: two algorithms provide bearing information, three provide TDOA data, and several provide target paramters. This data, with its associated error statistics, will be integrated in subsequent processing stages.

Parameter-List Processing

Given time-difference-of-arrival information as input information, the traditional algorithm for determining location involves combining results from at least three sensors to determine the intersection of a set of non-linear curves (see figure 4). The computational burden of such an approach is extremely high. With a sufficient number of sensors, however, it is possible to determine the target location using linear equations. This significantly reduces the computational burden, simplifies the problem of distributing the algorithm, and provides for graceful degradation as sensors fail. The largely open questions concern the behavior of these methods in the presence of noise, finding effective organizations for distributing these algorithms while retaining robustness, and determining the communication bandwidth requirements. Our approach has the potential of producing algorithms that have very favorable answers to these questions, (see Section 6.)

Extended Kalman filters are another class of parameter-list processing algorithms; a basically non-linear problem is continually re-linearized, with all unknown parameters included in the state. Although these methods are very popular in current centralized approaches, cooperatively distributed forms with reasonable communication bandwidths have been difficult to obtain. At the moment we can only suggest independently distributed approaches that are based on efficient exchanges of data, coupled perhaps with "event" detectors.

The difficulties with this approach arise because the linearization of the underlying nonlinear model and measurement equations obscures the structure of the generally well understood underlying physical processes. Furthermore, the nonlinearities make a performance analysis of these distributed filters very hard to obtain.

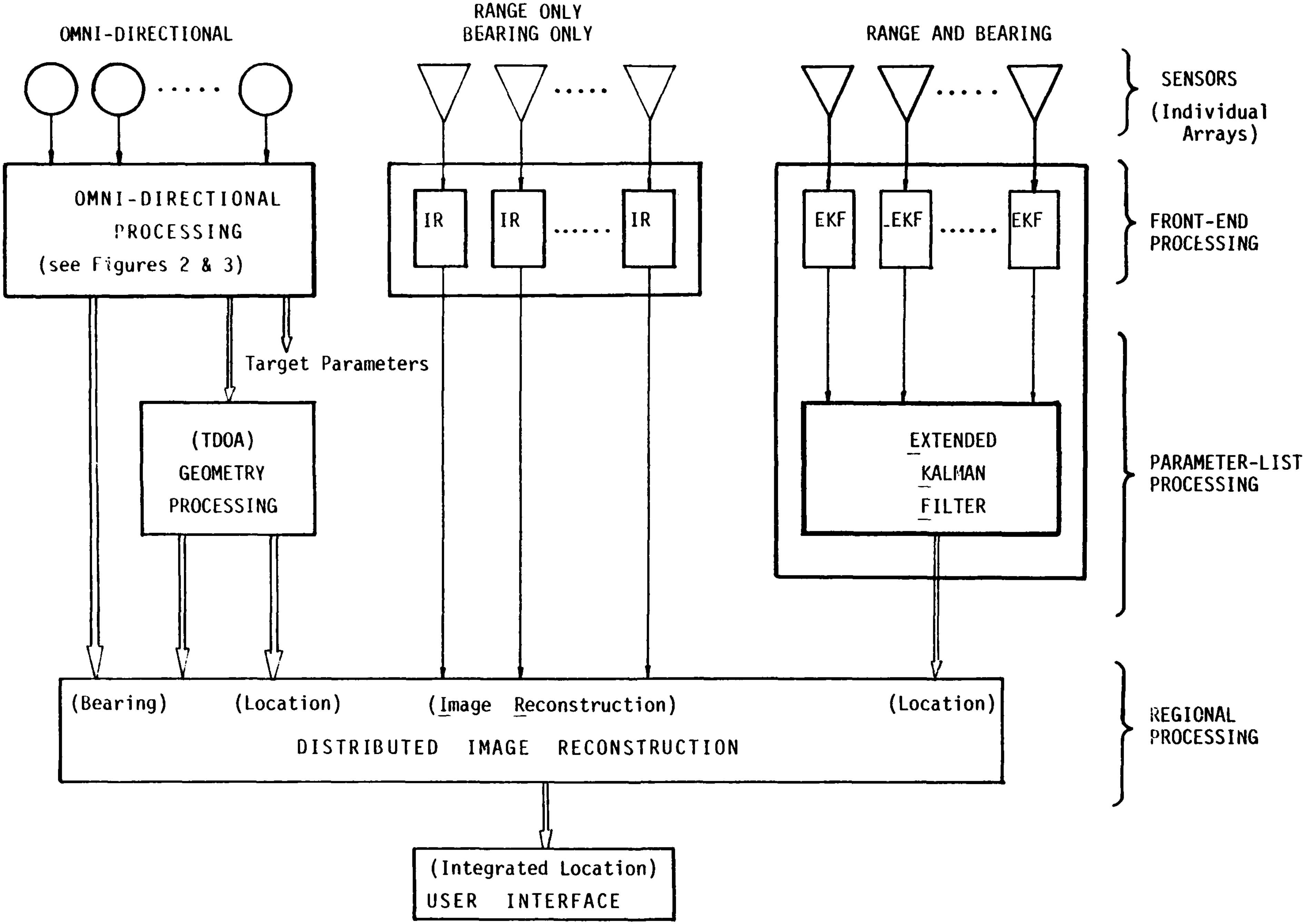


Figure 4. DSN System Structure

Regional Processing

Given information such as bearing-only or range-only data, back-projection is a well known technique for integrating this partial data into a single framework that will localize the target position. Intrinsically, this stage of data integration requires that each processing site have data from a large number of adjacent sites, i.e. those other sites whose sensors overlap the site in question. At this level, however, the communication requirements are typically reduced compared to earlier stages, although the computational burden may be very large.

The important enhancement we propose for this stage introduces error assessments into the back-projection process. With most sensors, some measurements have a greater confidence associated with them than others; this information should be incorporated, especially when using disparate sensor types or redundant measurements. The thesis of S. Wood [Wood] has analyzed the applicability and performance of state estimators to this problem. She demonstrated that significant performance improvements can be achieved over naive techniques, especially for ill-determined, constrained, or high signal-to-noise-ratio data.

Summary

An outline of possible component blocks for a distributed sensor network with cooperatively distributed processes has been shown in figure 4. Various constituent modules have been proposed; the choice of which blocks to implement in a given system depends upon the sensors available and the characteristics of the data. Several modules redundantly compute, for example, bearing or TOA information; only from an analysis of specific applications can we select the optimal configuration.

3. Signal Processing Modules for DSN

This section presents a brief summary of the types of algorithms we have investigated, the results obtained, and the various questions that still need to be answered. The approaches we have used can be classified into the following categories:

- Real Time Source Modeling (System Identification, ARMA)
    - Constant Source Parameter Models (CSPM)
    - Matrix Fraction Description Models (MFD)
    - Ladder Forms for "Event" Detection, (LLL)

- Multisite Lateration (MSL)

- Distributed State Estimation and Hypothesis Testing (DSE)
    - Distributed Information Filters (DIF)
    - Image Reconstruction Techniques (IRT)

- Two and Three Dimensional Signal Processing (2&3-D Systems)
    - Partial Differential Equation Models (PDE)
    - Delay Differential Equation Models (DDE)

3.1. Real Time Source Modeling

We are investigating a new approach to target bearing (or time-of-arrival) estimation by using system identification techniques. In this approach the parameters of an auto-regressive moving average (ARMA) model are chosen for best fit to the observed time series. If the source is purely autoregressive it can be shown that the moving average part of the model contains direct information about the TDOA, while the auto-regressive part represents the dynamics of the source process. Thus, identifying a system model provides information about the source bearing and about its spectrum.

Following this observation, we developed a new estimation scheme for TDOA which has a number of potential advantages over the more conventional estimator structures:

- No prior knowledge of the noise spectrum is required, since the proposed scheme automatically performs spectral estimation.

- By adaptively varying the ARMA coefficients to minimize an error criterion this estimator can handle nonstationary source processes (time-varying statistics) or moving sources (time-varying TDOA)

- Efficient recursive algorithms are available for computing ARMA coefficients for vector time series.

In summary, we applied system identification techniques to obtain a computationally efficient solution to the TDOA estimation problem, using our "fast algorithms". Other techniques for identifying the ARMA coefficients can be applied: repeated least squares, generalized least squares, maximum likelihood, instrumental variables, etc.

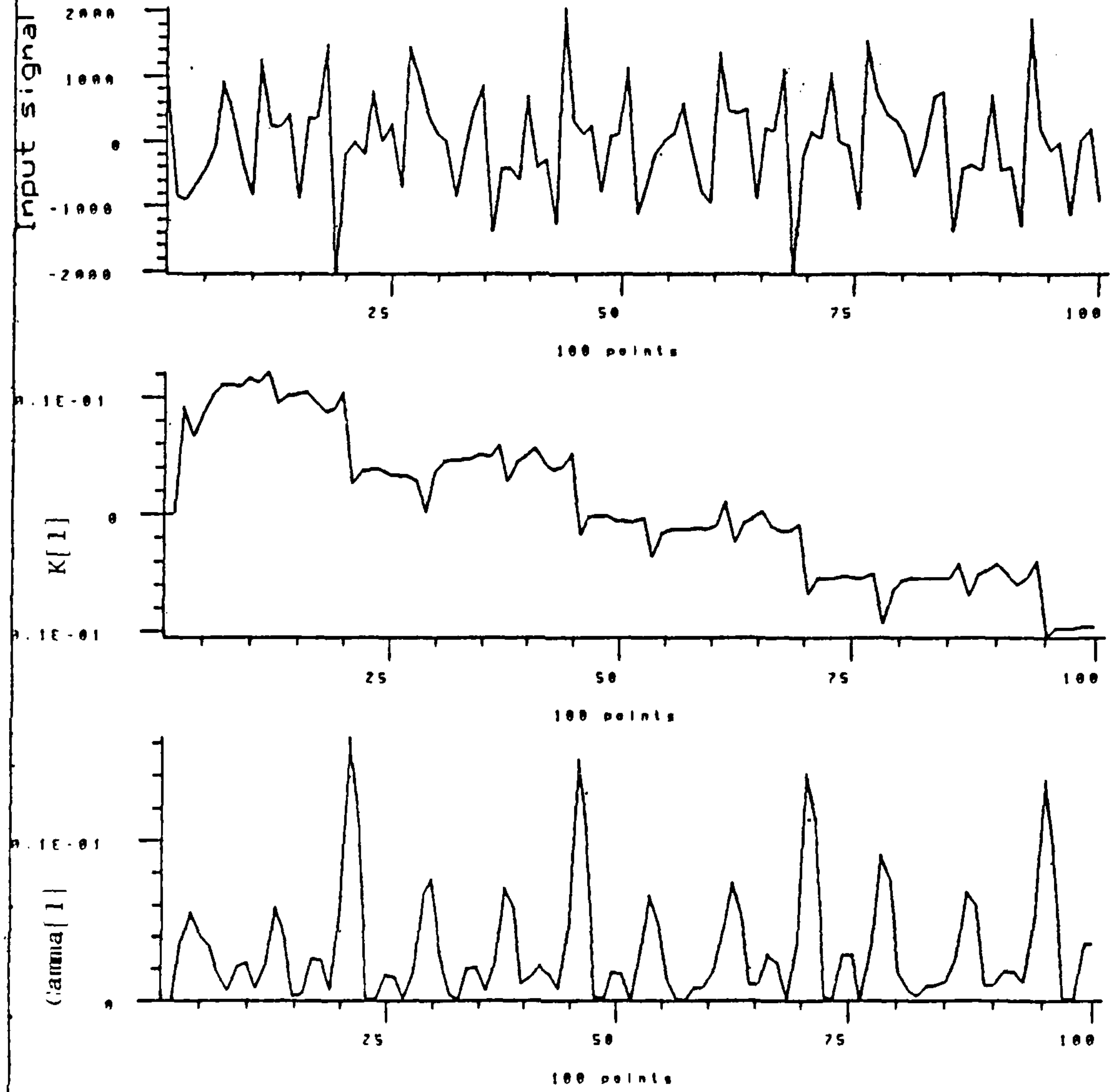
A major challenge of this approach is to obtain distributed forms of these algorithms, because of the inherent high complexity and tight coupling within these algorithms. One of the more promising methods we are exploring is based on information filter forms and "Matrix Fraction Descriptions" (MFD's), a dual representation of ARMA models. These forms can take better advantage of the inherent physical structure of the source location and characterization problem, e.g. the modeling of the "cascade" of source, propagation medium and receivers. This combination can actually more accurately be described by delay-differential equations, also a topic under investigation.

One potentially very important feature of these alternative forms is that their required communication bandwidths will not be a function of the number of sensors involved in a DSN but rather the number of sources. This is due to the fact that these models can separate the source and propagation models effectively. An other feature is the possible trade-offs, such as storage versus communication. For example, if in the *source spectral* information can be made available to the sensors and estimates are required only relatively infrequently, the number of variables communicated in the network is again neither a function of the number of sensors nor of the amount of data gathered at the sensors. In other words the sensor can effectively concentrate the received data, only communicating data that is of interest to the user. Section 6 describes some of the forms of distributed algorithms required for this application. The distributed algorithm for this case would in addition take advantage of Matrix Fraction Descriptions, that can effectively be used to provide *beam-forming* and subsequent optimal filtering.

A detailed description of our results in this area can be found in section 4 of this report: "A System Identification Approach to Locating Spatially Distributed Targets."

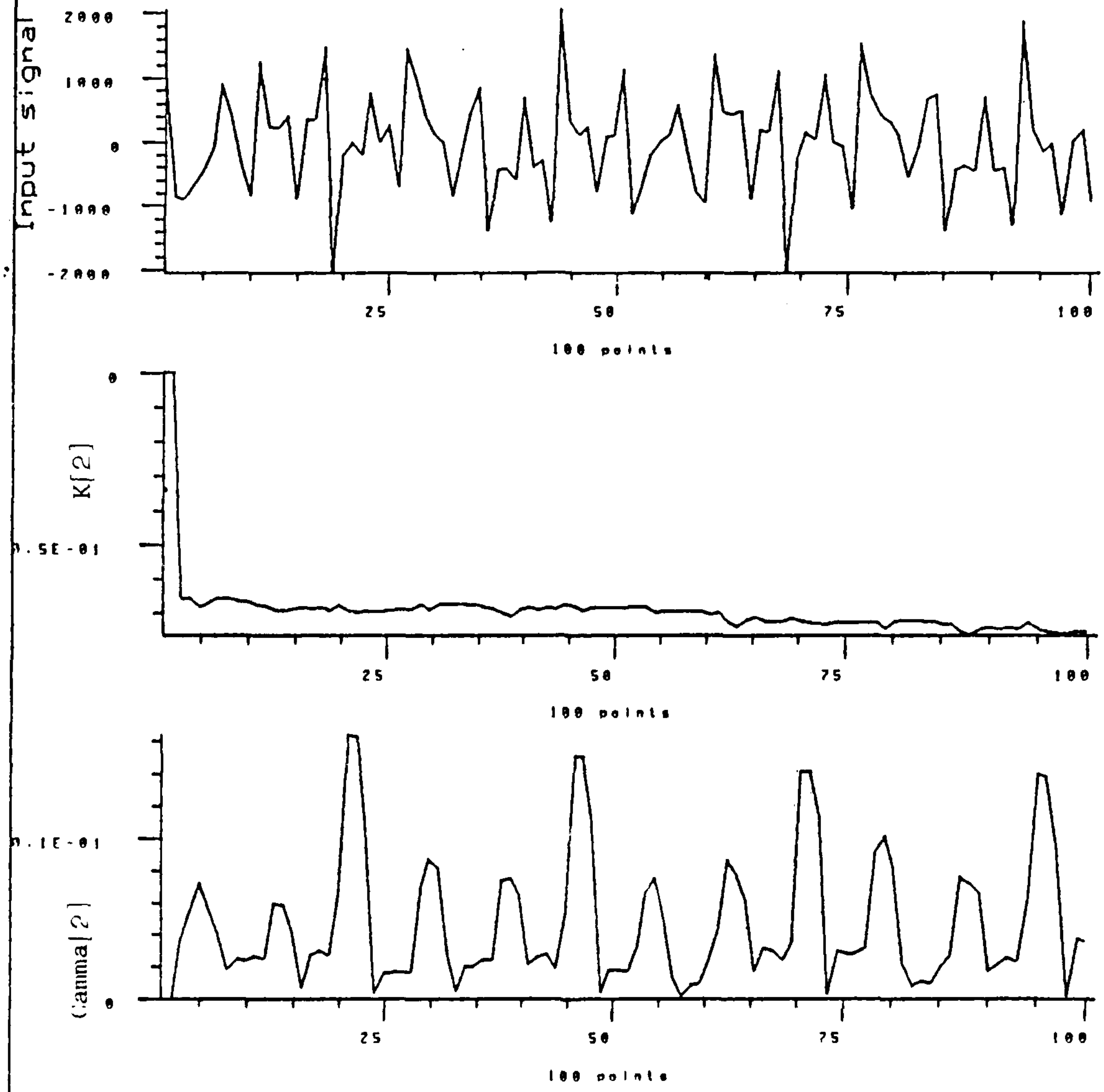
Ladder Forms for Log-Likelihood Detection

In a separate report on "Fast Parameter Tracking via Ladder Forms" [ML] we are reporting on the development and applications of ladder forms for on-line parameter estimation and tracking for single and multi-channel data. One of the major "discoveries" made in this report is the appearance of certain likelihood variables as optimal stepsizes in these recursive algorithms. Their extremely interesting behavior to jump components or "outliers" in the data create an interest in these variables in their own right, as well as in the many potential applications. An innovative approach to handling inordinate amounts of sensor data, these algorithms can identify significant "events" in the data (e.g. any discontinuity in a source, such as a change of course of a target under track). These discontinuities effectively provide a "sample-and-hold" capability to a parameter tracking algorithm, hence limiting the necessary processing effort to the neighborhood of these events, and providing a "time marking" capability of signals (useful for TOA estimation). In addition ladder forms are also very useful in data compression in order to reduce storage and communication requirements. Some of these applications were explored in the context of speech and other signals, see also [ML]. An example of processed acoustic data from Lincoln Labs (Lincoln [78]) is given in figure 5, where the "original" seismic signal during the closest approach of a fly-by, the i-th reflection coefficient $K(i)$ and the likelihood variable $gamma(i)$ is displayed. We note that even though the data was highly processed, the likelihood variable clearly shows distinct impulse like components or "events". Also note the change in the signum of $K(1)$, a sign for the change in doppler frequency from positive to negative. These examples are clearly preliminary results, but they show the promise of our approach.



Data: A7-jet; Receiver: geophone, SAV;
Samples: 1926-2025; Time: 6/27/77 5:36:30.00;

Figure 5a.



Data: A7-jet; Receiver: geophone, SAV;
Samples: 1926-2025; Time: 6/27/77 5:36:30.00;

Figure 5b.

3.2. Multi-Site Lateration (MSL)

Progress has been made on a solution to the multisite lateration problem using only time-of-arrival measurements which require *only* solutions to *linear* equations. This possibility was announced at an earlier site visit, and has now been confirmed (see also [Schmidt]). The significance of demonstrating the linearity of these equations arises because linearity is a sufficient condition for a distributed form of the algorithm. Any subset of sensors can optimally combine estimates of source/sensor coordinates, and these estimates can be combined globally using a variety of hierarchical organizations. The advantage of the linear equations is their inherent *fail-soft* property from N down to 4 stations, i.e., we have an "automatic re-configuration", since delayed or even missing measurement beyond at least 4 independent measurements have no catastrophic failure effects. The additional or missing measurements only raise or lower the confidence in the estimates to some degree. (Even the 4 station (linear) solution has at least a *single non-linear fail-soft* property, i.e. nonlinear solutions can be used *if one link fails.*) DSN systems based on "minimal" but nonlinear 3 independent station solutions however can *not fail-soft*! A single measurement or link failure can give catastrophic degradations i.e. a breakdown of the whole system in the worst case, since no redundancy would be present, a case that should definitely be avoided!

It is perhaps surprising, but not hard to prove by verification, that a linear set of equations can be given for the Euclidean coordinates of a source given range or range-difference measurements (or equivalently TOA or TDOA measurements.) The fact that the multi-site lateration problem (a set of quadratic equations) can be reformulated as a problem of solving linear equations, is reminiscent of a similar situation in least-squares estimation, where solutions to quadratic equations (quadratic cost) can be obtained solving linear equations (using orthogonal transformations). The multi-site lateration problem has the flavor of an inverse least-squares problem. A detailed description of one of our approaches to this problem can be found in section 6 of this report " Source Location from Time Difference of Arrival: A Linear Equation Approach."

3.3. Distributed State Estimation and Hypothesis Testing

Since many of the estimation schemes are based on state estimation formulation (e.g. the Kalman Filter), it is important to investigate distributed versions of state estimators. A simple form of a distributed information filter was presented in the appendix of our first DSN report. Various attempts have been made in the past to distribute the Kalman Filter. However, its inherent complexity and the high interaction between variables makes this virtually impossible, in the sense that either a lot of communication will be necessary (in a cooperatively distributed approach) or many variables have to be recomputed locally (in an independently distributed approach). The information filter form appears at the moment to be better adapted to distributed computations than the covariance (or Kalman) filter equations. In some cases, such a brute force implementation or distribution might be unavoidable. However, there are interesting cases where efficient distribution of the computations can be achieved by taking advantage of the structure of the problem. Triangular coupling of subsystems and degeneracies or non-existence of driving noises leads to efficiently distributed algorithms. Fixed sources or moving targets with piece-wise linear tracks are all examples of CSPM's.

An alternative solution is possible if global estimates rarely have to be computed, or local estimates have to be "synchronized" at a slower rate than the data that is gathered at the sensors. If these synchronization time points and the organization of sensors for obtaining a global estimate are agreed upon by the sensor sites by the time of the start of a new period, we can use our parallel processing algorithms, (see [Morf, Dobbins, Friedlander, Kailath], [Dobbins]) to develop essentially independently distributed algorithms that are communication bandwidth constrained. We can either fix a maximal deviation in the error-covariance of all estimates and determine the necessary synchronization intervals, or we can determine these intervals by fixing the communication bandwidth and thereby limiting the number of information exchanges necessary to achieve the global synchronization. The equations can be derived relatively easily using our scattering theory approach to least-squares estimation, see e.g. [Friedlander]. A detailed description can be found in section 6 of this report: "Distributed Algorithms for Estimation and Detection."

3.4. Two and Three Dimensional Signal Processing

Our work on two and three dimensional signal processing is characterized by two efforts.

a) A basic mathematical approach has been developed that is aimed at providing the proper mathematical tools such as state space models and transfer function or frequency domain representations. We have, in the meantime, obtained very basic results that may have a great potential impact on the problem of obtaining truly distributed solutions to two and three dimensional signal processing, see [Levy], [Levy,Morf]. However, their real significance is still hard to evaluate, partially because one of the basic results we are hoping to obtain is still missing: a demonstration of the existence of innovations representations for two and three dimensions.

One of our basic results shows that some of the structure of two dimensional problems are definitely different from three dimensional equivalents. For example, the idea of implementing filters in a cascade form, very basic in one dimension, generalizes (even for multiple channels) to two dimensions; however, this is *not true* in *three dimensions*. This result does not imply that solutions do not exist in three dimensions, it only points out that solutions in three dimensions can look quite different from the lower (one and two) dimension equivalents.

b) As discussed in the Tutorial Survey presented at CMU, one method for estimating source probability distributions is based on image reconstruction techniques, (IRT). The comparatively simple *back-projection* technique, (see e.g. [Lincoln Lab, CMU workshop],) does not, however, lead to the best estimate of the source distribution, but rather to what is known in the estimation literature as a special case of the information state, see e.g. [Wood]. More precisely, the information state is the product of the best estimate times the inverse of the error-covariance: a "smeared out" version of the best estimate. The best estimate can be obtained either by carrying along the covariance matrix (recursively) and multiplying the information state by the inverse, or by recursively computing the best estimate directly. We showed that the direct approach is equivalent in the continuous limit to the so-called "convolve and back project" solution, whereas the information filter approach is a "first back-project then convolve" solution. The latter currently appears to be more adapted to distributed computations; however, one of our earlier fundamental results showed that both approaches are linked via an inversion duality, hence ultimately we might also be able to find efficient distributed algorithms for the direct form. Both forms generally involve carrying along a form of the covariance matrix (a matrix with $n^2/2$ entries in general, where n is the number of variables to be estimated). The memory requirements for a brute-force implementation of these methods could, therefore, be prohibitive. One advantage, however, is that no limit on the number of targets is required, since a spatial discretization is used. This implies that targets are lumped together if they are within a single spatial "bin". Other advantages of this approach derive from the linearity of the optimal estimator (for Gaussian distributions), from flexible distributing computations, robustness of the source distribution estimates, and "automatic reconfiguration" after sensor losses. We are still studying the implications of this approach on computational and communication requirements, as well as other system parameters. Our recent results in 2-D systems [Levy,Morf] are expected to play an important role in this context, in particular in displaying and exploiting sparseness of matrices for computational and other benefits.

3.4.1 Delay Differential Equation Models

Since sources emit signals that generally have a line spectrum and that propagate (i.e. are delayed) through a medium, we can conclude that natural mathematical models are systems of delay-differential systems. B. Levy was able to show in his thesis that in the generic case at least two sensors are required in order to be able to observe (or two inputs in order to control) a delay-differential system.

This implies that a (generally moving) source can be located from appropriate measurements of only two sensors, e.g. time and frequency of arrival type measurements of two sensors can be sufficient for estimating the trajectory of a source! This is a very fundamental mathematical result; its full impact on the DSN problem remains to be determined.

We may note that our other approaches that are based on ARMA and MFD models essentially make use of discretized delay-differential equations, hence their structure is inherited by the discretized equivalents.

A System Identification Approach to Locating Spatially Distributed Sources

B. Friedlander*, M. Morf**

Abstract

A new framework is presented for treating the problem of estimating the location of a multiple of point sources from multi-sensor measurements. Estimating the source location is formulated as a system identification problem for the system consisting of the source and propagation models. It is shown how the time difference of arrival (TDOA) can be found from the system parameters. Features of the proposed approach include: simultaneous estimation of multiple sources, capability of handling multipath propagation and nonstationarity of the source and noise processes, and simultaneous estimation of the source location and its spectral "signature". The basic concepts are described in detail and some sample algorithms are presented. A number of interesting connections between the source location problem and system theoretic issues are presented.

* Systems Control Inc., Palo Alto, CA 94304.

**Information Systems Laboratory, Stanford University, Stanford, CA 94305. This work was supported by the Defense Advanced Research Projects Agency through contract MDA 903-78-C-0179, and the facilities at the Stanford Artificial Intelligence Laboratory.

I. Introduction

In many practical problems it is necessary to determine the location of signal (and noise) sources from measurements provided by one or more sensors. Typical applications include:

- Acoustic surveillance systems (e.g., detection of low flying aircraft);
- Seismic arrays for seismic exploration, monitoring earthquakes and nuclear explosions, or detecting vehicle movements;
- Antenna arrays for radio astronomy or electronic surveillance (e.g., direction finding);
- Multiple radar systems for detection and tracking.

In the simplest case the sensor outputs consist of amplitude scaled and delayed replicas of the waveform from the distant source, corrupted by additive noise. One of the most widely used methods of estimating the location of the source is by finding the time-difference-of-arrival (TDOA) of the propagating signal to the different sensors. It has been shown [1-2] that all the information about the source location (range and bearing) is contained in these time-of-arrival differences (TDOA's), or equivalently, in the difference of the sensor/source range for a pair of sensors. In other words, estimating the source location is equivalent in some sense to estimating TDOA's for the various sensors.

Methods for estimating TDOA, such as maximum likelihood estimators, were developed by several authors [3-4]. These estimators are usually implemented by using one of the following structures:

(i) A beamformer (with appropriate pre-filtering) followed by a square-law device and an integrator.

(ii) Pre-filtering followed by a multiplier correlator.

The computations are often performed using Fourier transforms, but time-domain implementations are also quite common. Various suboptimal TDOA estimators have also been derived [4]. The TDOA estimator is by no means the only method for estimating source location, but it provides a convenient prototype for our discussion. For details about other approaches, see [5-8].

These estimators have been successfully used on a variety of applications. There are, however, certain aspects of the currently used techniques which are somewhat unsatisfactory. For example:

- Most of the estimators are designed to be optimal for the single source case. When multiple sources are present they are considered one at a time (while other sources are thought of as "interference"). In general the treatment of multiple source proceeded in an 'ad hoc' fashion and there is a need for a more systematic approach.
- The performance of most estimators is degraded in the presence of multipath propagation.
- These approaches are highly centralized and do not seem to be easily adaptable to distributed processing. The current trend toward using multiple micro- and mini-computers rather than a single powerful central computer makes it very advisable to develop algorithms suitable for distributed processing. In addition to cost effectiveness, distributed systems have

other potential advantages like speed and high reliability (survivability).

These and other consideration provide motivation for searching alternative approaches to the source location problem.

In this paper we introduce a new framework for estimating TDOA's or source location. The objective of this paper is to outline our approach and indicate some interesting connections between the source location problem and system theoretic issues. We limited ourselves here to a general discussion of the issues and deferred many of the details to later papers.

The basic idea is to use system identification techniques to estimate the parameters of the system comprised of the source (or sources) and the sensors. The TDOA estimates can then be easily found (even for multiple sources) from these system parameters as will be shown in the next section.

The proposed approach leads to new estimator structures, quite different from the classical beamformer or cross-correlator. This new estimator has a number of desirable features including:

(i) Estimating multiple-source locations.

The conventional TDOA estimators treat a single source at a time and treat any other sources as noise or interference. In Section III we will present an optimal estimator for multiple-sources.

(ii) Simultaneous estimation of source location and its spectrum.

In many applications (e.g. passive sonar) one wishes to estimate the source spectrum for recognizing or classifying the source type (i.e. finding its spectral "signature") or for other purposes. The process of spectral estimation is usually carried out separately from the source location estimation. In the proposed approach these two aspects of the problem can be handled simultaneously. This leads to the possibility of using a priori information about the source spectrum to enhance detection of specific source types while rejecting other source types. In other word, the approach has a certain built in discrimination capability.

(iii) Capability of handling multipath propagation.

In many cases the source signals find several different propagation paths (with different delays) to the sensor. These multipath signals are capable of "confusing" some of the TDOA estimators and degrading their performance. Insensitivity to multipath propagation is an inherent feature of our approach, as will be discussed in Section II.

(iv) Tracking moving sources and adaptivity.

Some implementations of our proposed approach (Appendix A and B) are capable of tracking (even rapidly) time-varying system parameters and therefore capable of tracking moving sources. Furthermore, these algorithms are capable of performing optimal estimation (e.g. in the least-squares or maximum likelihood sense) for nonstationary processes. This is a favorable property since in many applications the assumption of stationarity does not hold (even if the source is not moving). In fact, the adaptive nature of these algorithms makes possible optimal estimation without prior knowledge of the source or noise statistics.

The structure of this paper is as follows: in Section II the basic approach is described in detail for the case of a single autoregressive (AR) source. In Section III the approach is applied to multiple AR sources, and some of the difficulties involved are discussed. Section IV summarizes possible extensions of our approach to a more general class of sources (autoregressive moving-average, ARMA), to moving sources and to direct estimation of source location (rather than TDOA).

Finally, it should be noted that the emphasis in this paper is on the conceptual development of a unifying framework for processing multi-sensor multi-source data. We will not go into the details of deriving the actual algorithms (some of which are presented in the appendices) or into various numerical and other practical issues involved in the actual application of such algorithms. These will be treated in later papers.

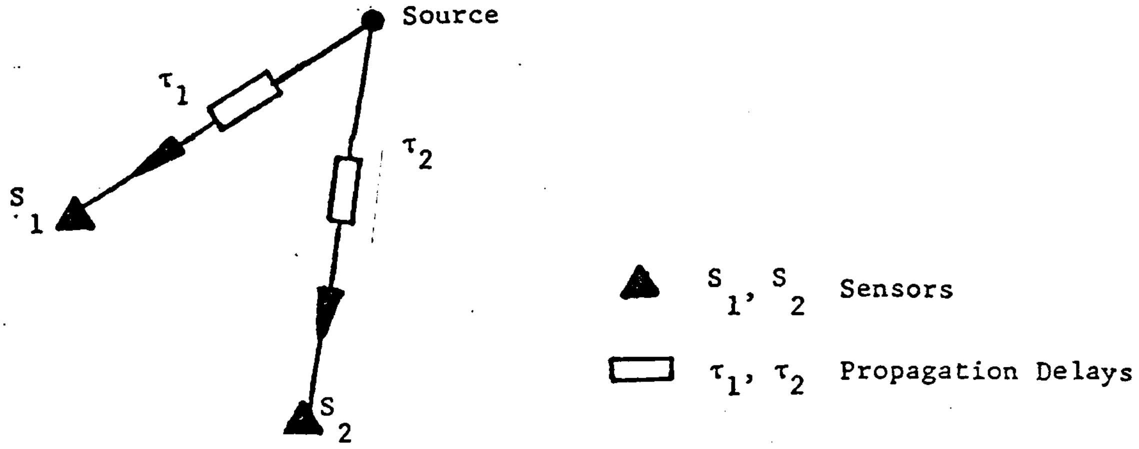


Fig. 1 Two sensors and one source.

II. A Single Autoregressive Source

To illustrate the basic ideas of our approach we start with the following simple problem as shown in Figure 1. Two sensors are measuring the signal propagating from a source located somewhere in the plane. We assume that the propagation involves only some time delays and attenuation. Thus, the outputs y_1, y_2 of the two sensors can be modeled as:

$$y_1(t) = x(t-\tau_1) + n_1(t) \tag{1a}$$

$$y_2(t) = c\, x(t-\tau_2) + n_2(t) \quad , \tag{1b}$$

where τ_1, τ_2 are the propagation delays from the source to the two sensors, c represents attenuation and n_1, n_2 are independent measurement noise processes. The time sampled version of these outputs will be written as:

$$y_1(k) = x(k-D_1) + n_1(k) \tag{2a}$$

$$y_2(k) = c\, x(k-D_2) + n_2(k) \quad , \tag{2b}$$

where

$$t = k\ \Delta t \ , \quad \tau_1 = D_1\, \Delta t \ , \quad \tau_2 = D_2\ \Delta t.$$

Note that the delays τ_1, τ_2 are assumed to be integer multiples of the sampling period. No difficulties arise when τ_1, τ_2 are non-integer multiples provided that the sampling period Δt is properly chosen. This point will be discussed in more detail later.

The source process $x(k)$ is an autoregressive process of order n, i.e.

$$x(k) = - \sum_{i=1}^{n} a_i\, x(k-i) + u(k) \quad , \tag{3}$$

where $u(k)$ is a white driving process. Taking the z-transform of equation (2) we get

$$Y_1(z) = z^{-D_1} X(z) + N_1(z) = z^{-D_1} \frac{U(z)}{a(z)} + N_1(z) \tag{4a}$$

$$Y_2(z) = c\, z^{-D_2} X(z) + N_2(z) = c\, z^{-D_2} \frac{U(z)}{a(z)} + N_2(z) \ , \tag{4b}$$

where

$$a(z) = 1 + \sum_{i=1}^{n} a_i z^{-i} \tag{5}$$

Written in vector form the transfer function from the driving process u to the sensor outputs y_1, y_2 is

$$Y(z) = B(z)\, X(z) + N(z) = B(z) \frac{1}{a(z)} U(z) + N(z) \ , \tag{6a}$$

where

$$B(z) = \begin{bmatrix} b_1(z) \\ b_2(z) \end{bmatrix} = \begin{bmatrix} z^{-D_1} \\ c\, z^{-D_2} \end{bmatrix} \tag{6b}$$

Note that the numerator of this transfer function contains the information about the target location (i.e. the TDOA), while the denominator contains the dynamics of the source process.

This simple example suggests the following method for estimating TDOA: fit an ARMA model to the observed time series $y(k)$. That is, find coefficients $\{ a_i, B_i \}$, where

$$y(k) = - \sum_{i=1}^{n} a_i\, y(k-i) + \sum_{i=1}^{n} B_i\, u(k-i) + e(k) \ , \tag{7a}$$

where $e(k)$ is a correlated noise process given by

$$e(k) = n(k) + \sum_{i=1} a_i\, n(k-i) \ . \tag{7b}$$

For simplicity, assume a large signal-to-noise ratio (SNR) in which case $e(k)$ is negligible. In other words, let

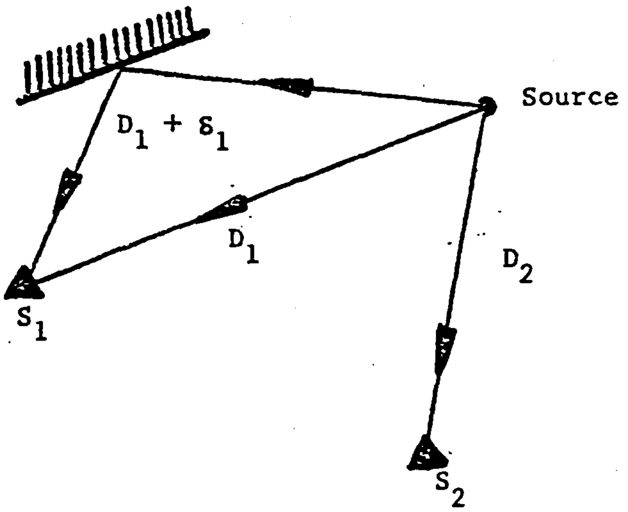


Fig. 2. Multipath propagation.

$$y(k) = -\sum_{i=1}^{n} a_i\, y(k-i) + \sum_{i=1}^{n} B_i\, u(k-i)\ . \qquad (8)$$

This assumption will be relaxed in section III.

The coefficients $\{ a_i,\ B_i \}$ will be chosen to best fit equation (7) in the mean square error sense, by performing system identification based on the observed outputs $y(k)$. Then, by examining the numerator coefficients B_i, the TDOA can be found. It should be noted that the numerator polynomial found in this manner will not be unique, since without having direct measurements of the source $x(k)$ the absolute delays (i.e. the degrees D_1, D_2 of the polynomials b_1, b_2) can not be determined. However, the difference in the degrees of the polynomials b_1, b_2 will be unique and equal to the TDOA, $\Delta_{12} = D_1 - D_2$, which provides the desired information about the source bearing.

Consider now the case where multipath propagation is present. For example, assume that sensor 1 received a delayed version of the direct signal, as indicated in Figure 2.

In this case

$$y_1(k) = x(k-D_1) + c_1\, x(k-D_1-\delta_1) + n_1(k) \qquad (9a)$$

$$y_2(k) = c_2\, x(k-D_2) + n_2(k) \quad , \qquad (9b)$$

where c_1 represents the attenuation 'he indirect propagation path and δ_1 its additional delay. The numerator polynomials in this case will be

$$b_1(z) = z^{-D_1} + c_1\, z^{-(D_1 + \delta_1)} \qquad (10a)$$

$$b_2(z) = c_2\, z^{-D_2} \quad . \qquad (10b)$$

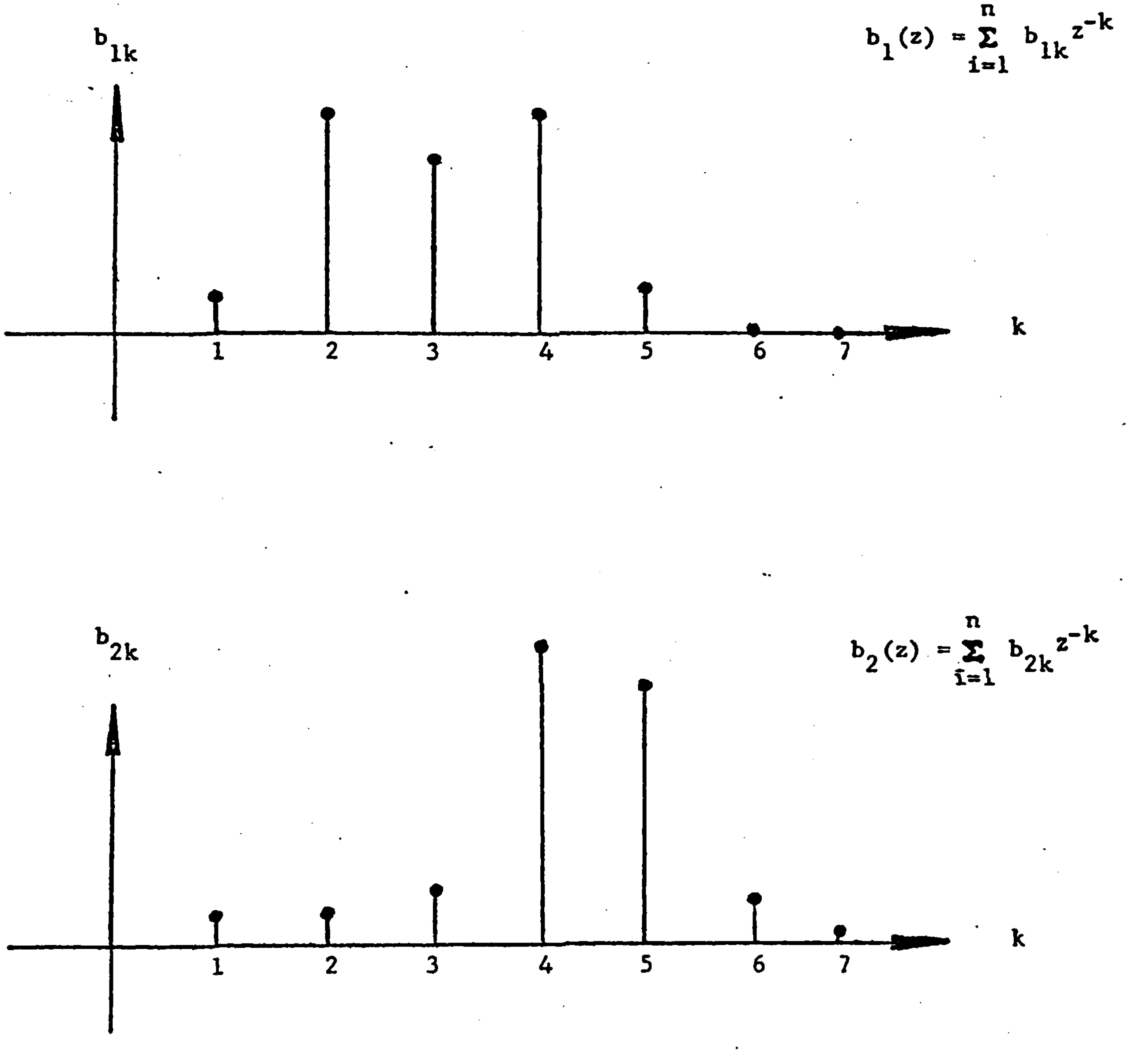


Fig. 3. Numerator coefficients.

Thus, the multipath propagation introduces additional terms in the numerator polynomials. Note, however, that these additional terms will always have a higher degree than the term corresponding to direct propagation, because of the additional delays involved. Therefore, if we look at the degrees of the first nonzero coefficient in $b_1(z)$, $b_2(z)$ (i.e. the coefficient of the lowest degree term) we can still find the TDOA, $\Delta_{12} = D_1 - D_2$!

To illustrate this procedure, assume that we performed the ARMA fitting on a given data set $\{y(k)\}$ and found the numerator coefficients. As plotted in Fig. 3 the first significant non-zero coefficient of $b_1(z)$ is $b_{1,2}$, and the first non-zero coefficient of $b_2(z)$ is $b_{2,4}$. Thus, the TOA difference in this case is $4 - 2 = 2$. The small nonzero coefficients are due to imperfect modeling because of measurement noise, while the fairly large coefficients following $b_{1,2}$, $b_{2,4}$ are due to multipath propagation.

Figure 3 also provides an indication of what will happen if the delays are non-integer multiples of the sampling period: instead of having a single nonzero coefficient associated with a given delay, we will have two large coefficients whose relative magnitudes reflect how close the real delay is to the delay represented by that coefficient. (For example, if $k \Delta t \leq \tau_1 \leq (k+1) \Delta t$ we may expect both $b_{1,k}$ and $b_{1,(k+1)}$ to be nonzero). In other words, as long as the sampling rates are sufficiently high compared to the bandwidth of the underlying process, the same approach will work for non-integer time-delays.

This approach can, of course, be easily generalized to the case when there are more sensors. In the case of M sensors we have

$$B(z) = [\ b_1(z),\ b_2(z),\ .\ .\ .\ .\ ,\ b_M(z)\ ]^T\ . \qquad (11)$$

Comparison of the degrees of the first nonzero terms of $b_i(z)$ and $b_j(z)$ will give the TDOA, $\Delta_{ij} = D_i - D_j$. The generalization to the case where more than one source is present is more complex and will be discussed in Section III.

Identification Techniques

The discussion so far has illustrated that infromation about the source bearing, i.e. TDOA, is contained in the coefficients B_i of the ARMA representation of the observed measurements. The question remains, how to compute these ARMA coefficients $(a_i,\ B_i)$ given a set of measurements $\{ y(k) \}_{k=0,K}$.

This type of problem has been widely studied in the context of estimation and identification of linear systems. Finding $\{ a_i,\ B_i \}$ in equation (8) is usually referred to as "the case of correlated residuals". A simpler version of this problem is the following:

Assume that the measurements $y(k)$ are the output of an AR system driven by white noise $v(k)$

$$y(k) = - \sum_i a_i\ y(k-i) + v(k)\ . \qquad (12)$$

Find a_i which will minimize the mean square error $E\{ [y(k) - \hat{y}(k)]^2 \}$ where $\hat{y}(k) = - \sum_i a_i\ y(k-i)$. The solution of this problem is fairly straightforward and typical algorithms can be found in [9-10].

As can be seen by comparing equations (8) and (12), in the source location problem $v(k)$ is a correlated sequence of random variables. In fact it has the form

$$v(k) = \sum_{i=1}^{n} B_i\, u(k) \tag{13}$$

where $u(k)$ is an independent white noise process.

Identifying the system when the sequence of $v(k)$ (also called the residuals) are correlated is somewhat more complicated. Several techniques have been suggested to deal with this case, including:

(a) repeated least-squares

(b) generalized least squares

(c) the maximum likelihood method

(d) instrumental variables

For references to these techniques see the survey by Astrom [9].

More recently a new approach has been developed by Morf [11-13] which provides efficient forms of the so-called exact recursive least squares algorithms, i.e. RML 1 or 2 (see e.g. [14], [15]). These new forms have the added advantages of computational efficiency and fast parameter tracking capability [16-17]. The last property is important for tracking moving sources, since the ARMA model corresponding to such sources has time varying parameters. We will discuss this further in section IV.

Another attractive feature of the exact recursive least-squares algorithms is that they do not require stationarity of the the underlying source and noise process. This is in contrast to the type of algorithms which are used in conventional array processing where stationarity is often a necessary assumption. Furthermore, these algorithms preserve their computational efficiency under nonstationarity, while other approaches lead to solutions of increased complexity when stationarity is lost (see e.g. [20-21]).

The details of these (and other) algorithms can be found in the references and the appendices. It should be emphasized that the significance of the problem formulation presented earlier (equation (8)) is that it makes it possible to apply any identification technique to the source location problem. The algorithms presented in the appendices are merely examples. A comparative evaluation of different identification algorithms in the context of the source location problem has not yet been performed. However, it is our belief at this point that the exact recursive least squares algorithms are particularly suitable for this application. Further discussion of these algorithms and simulation results will be presented in future papers.

The identification techniques mentioned above provide estimates of $\{ a_i, B_i \}$. As mentioned in the introduction the coefficients a_i are related to the spectral "signature" of the source. In fact the power spectral density of the source is given by

$$R(z) = \frac{1}{a(z)\, a(1/z)} \quad . \tag{14}$$

Estimating the spectrum by autoregressive modeling is considered one of the better spectral estimation techniques (see e.g. [18-19], popularly referred to as "maximum entropy" spectral analysis). We therefore have reason to expect good performance from our proposed approach.

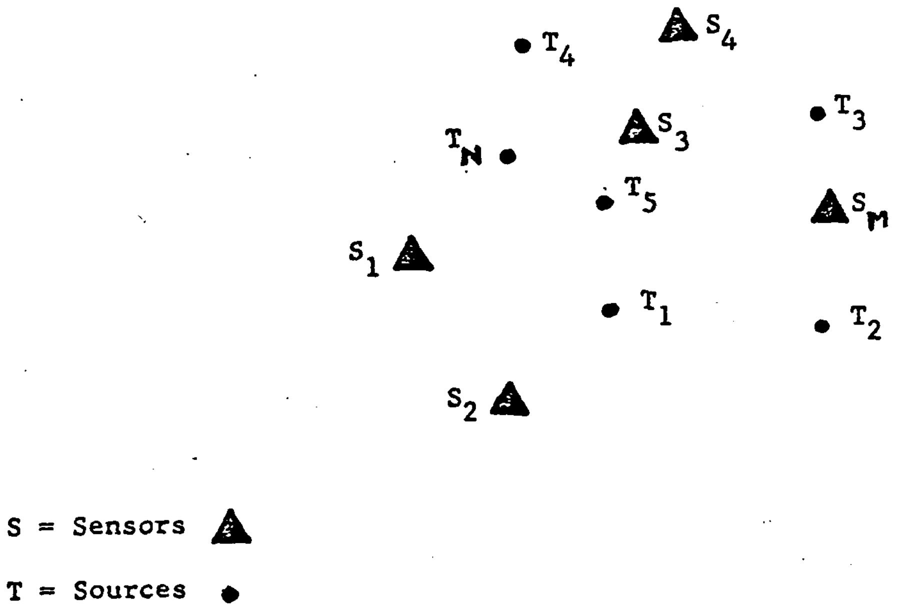


Fig. 4. N sources and M sensors

III. Multiple AR Sources

In many applications the signals received by the sensors are generated by several sources. Sometimes we are interested in all of these sources, while in other situations only a subset of them is of interest; the others are merely "interference". For example, suppose we want to detect low flying aircraft, but the sensors also receive noise from a nearby generator or truck. The method described here is capable of simultaneously estimating the bearing (TDOA) of all sources in the area covered by the sensors. Thus, the existence of point interference sources does not represent any special problems in this approach. The interference is simply another source whose parameters are to be estimated. (A good example of "relabeling" sources appears in the estimation theory explanation of "noise cancelling" discussed in [17]). In the following discussion we will, therefore, not distinguish between interferers and sources of interest.

Consider an area containing N sources and M sensors (Fig. 4). Each source is assumed to be of an autoregressive type, i.e.

$$x_i(k) = -\sum_{j=1}^{n_i} a_{ij}\, x_i(k-j) + u_i(k) \quad , \quad i = 1, \ldots , N \tag{15a}$$

or written in vector form

$$x(k) = -\sum_{j=1}^{n} A_j\, x(k-j) + u(k) \quad , \tag{15b}$$

where

$$n = max\, \{n_i\}$$

$$x(k) = [\, x_1(k), \;\ldots\; , \; x_N(k)]^T$$

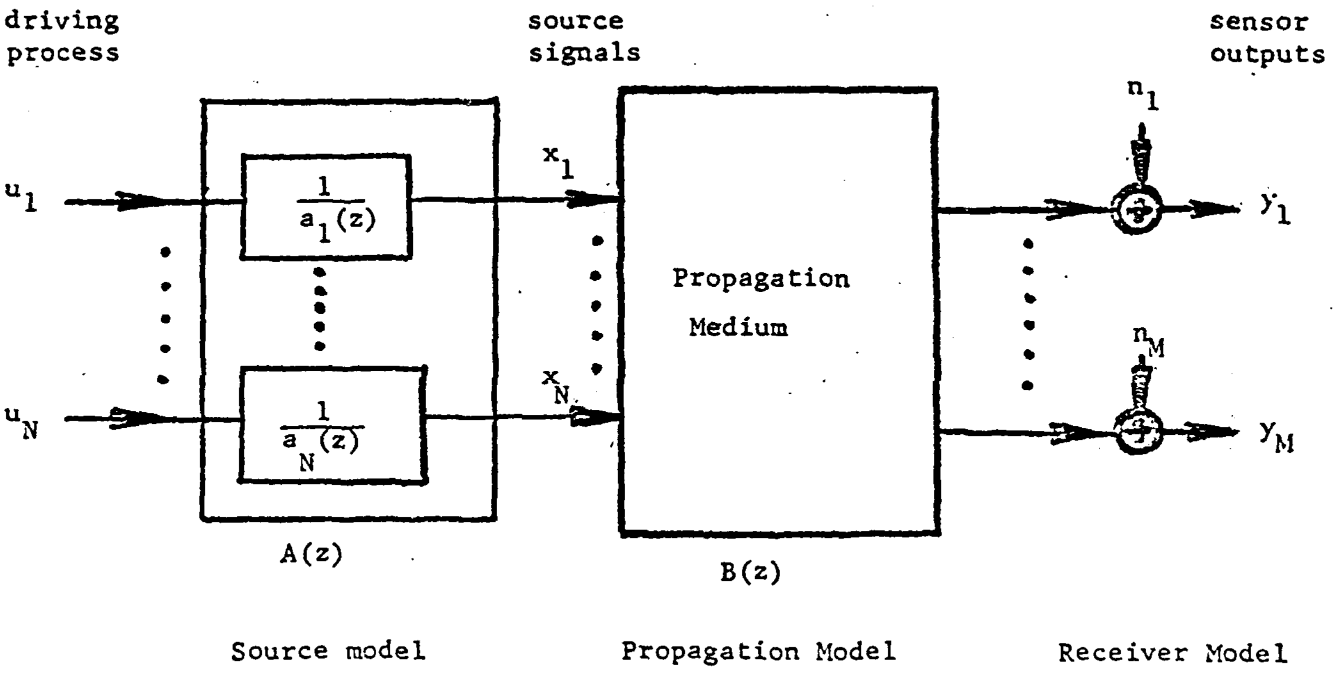


Fig. 5. Source model with propagation medium and receiver model.

$$A_j = diag\{ a_{ij} \} .$$

Written in the z-transform domain we get

$$X(z) = A^{-1}(z)\, U(z) \quad , \qquad (16a)$$

where

$$A(z) = \mathrm{I} + \sum_{j=1}^{n} A_j\, z^{-j} = diag\{ A_i(z) \} \quad , \qquad (16b)$$

$$A_i(z) = 1 + \sum_{j=1}^{n_i} a_{ij}\; z^{-j} \quad . \qquad (16c)$$

In the previous section we have seen that the propagation from a single source to the M sensors can be represented by a polynomial matrix of dimension M x 1. It is easy to see that in the multi-source case we can write

$$Y(z) = B(z)\, X(z) + N(z) = B(z)\, A^{-1}(z)\, U(z) + N(z) \quad , \qquad (17)$$

where $B(z)$ is an M x N polynomial matrix (and $X(z)$ is an N x 1 vector of polynomials). Each column of this matrix represents the transfer function from one of the sources to the sensors.

Using the same reasoning as before, we may conclude that the *i-th* column of $B(z)$ provides information about the TDOA for signals generated by the *i-th* source. Thus, estimating the coefficients of $B(z)$ will essentially solve the multi-source location problem. Estimating $A(z)$ will, of course, provide information about the spectrum of each source.

The problem of locating multiple sources and finding their "spectral signatures" can, therefore, be formulated as an identification problem of a multi-input (N) multi-output (M) system, see also Fig. 5. All the techniques which have been developed for system identification can, at least in principle, be applied to solve this problem. There are, however, a number of difficulties in the multi-source case, which will be discussed next.

Matrix Fraction Descriptions (MFD)

In equation (17) we have derived a transfer function of the form

$$H(z) = B(z)\,A^{-1}(z) \quad . \tag{18}$$

This type of system representation is called the Matrix Fraction Description (MFD) of a multi-input multi-output linear system. The properties of MFD have been extensively treated in the literature of linear systems, e.g. [22-23]. Here we shall only mention some of the basic facts which are relevant to the source location problem.

(i) Minimal Realizations and Uniqueness

The MFD representation in equation (18) is clearly nonunique. If $T(z)$ is any polynomial matrix, we can write

$$H(z) = B(z)\,A^{-1}(z) = (\,B(z)\,T(z)\,)\;(\,A(z)\,T(z)\,)^{-1} \tag{19}$$

i.e. the numerator and denominator matrices can have arbitrary "common factors", called common divisors in the matrix case. It is useful, therefore, to restrict our attention to the so-called minimal realization of the system $H(z)$, which is a pair of matrices $A(z)$, $B(z)$ which have no more "common factors" and which have in some sense the lowest possible "order". It turns out that the minimal representation of the system $H(z)$ is still nonunique. The nonuniqueness is due to the fact that a minimal representation $\{A(z),\ B(z)\}$ can be multiplied by any unimodular matrix U and remain minimal (i.e. $\{A(z)U,\ B(z)U\}$ is still a minima! ~epresentation). Unimodular matrices are polynomial matrices whose inverses are also polynomial. They are the analog of a constant multiplier in the single-input single-output case where a transfer function $b(z)/a(z)$ can be uniquely determined only up to multiplication by a constant c, *i.e.* $b(z)/a(z) = c\, b(z)/ca(z)$.

Nonuniqueness of this type may "mix up" the rows and columns of $B(z)$, making it difficult to extract the TDOA information. If we assume the sources to be independent, the denominator $A(z)$ in the source location problem has a very special structure, namely it is a diagonal matrix. Therefore, if we will choose that particular minimal realization which has a diagonal denominator matrix, we can get a unique MFD representation of the source/sensor system (under some weak assumptions which will be met in all the applications considered here). In other words, the special structure of the source location problem can be exploited to define a unique representation of the system.

(ii) Left and Right MFD's

The form of the transfer function as given in equation (18) is only one of two basic forms of MFD's. An alternative representation is

$$H(z) = \hat{A}^{-1}(z)\, \hat{B}(z) \ , \qquad (20)$$

where $\hat{A}(z)$ is an $M \times M$ polynomial matrix and $\hat{B}(z)$ is $M \times N$. ($A(z)$, $B(z)$) and ($\hat{A}(z)$, $\hat{B}(z)$) are called Right MFD and Left MFD, respectively.

Note that the formulation of the source location problem leads to a Right MFD. Unfortunately, the process of identifying an ARMA model leads naturally to a Left MFD. To see this consider a multi-input multi-output ARMA model of the form,

$$y(k) = -\sum_{i=1}^{n} \hat{A}_i \, y(k-i) + \sum_{i=1}^{n} \hat{B}_i \, u(k-i) \ . \tag{21}$$

Taking z-transforms will give

$$\hat{A}(z)\, Y(z) = \hat{B}(z)\, U(z) \tag{22}$$

or

$$Y(z) = \hat{A}^{-1}(z)\, \hat{B}(z)\, U(z) \ , \tag{23}$$

where

$$\hat{A}(z) = I + \sum_{i=1}^{n} \hat{A}_i \, z^{-i} \ , \tag{24}$$

and

$$\hat{B}(z) = \sum_{i=1}^{n} \hat{B}_i \, z^{-i} \ . \tag{25}$$

Currently available identification techniques provide ways for estimating ARMA coefficients and will therefore give estimates of the Left MFD ($\hat{A}(z)$, $\hat{B}(z)$). In order to find the TDOA, we will first have to find the Right MFD of the system ($A(z)$, $B(z)$), given estimates of the Left MFD. Various techniques for going from Right- to Left MFD's are available. Recently, some algorithms with good numerical properties have been reported. Note, however, that in this case we want to go from a Left MFD to a Right MFD with a specific structure (namely, diagonal A(z)). The question of how to do this in a reliable and efficient way is not completely resolved at this time.

A more direct approach would be the development of estimation algorithms for ($\hat{A}(z)$, $\hat{B}(z)$) directly, i.e. identifying the Left MFD. No simple algorithms seem to be available and further work is needed in this area.

Most of the difficulties mentioned here did not arise in the single-source case. This was due to the fact that in that case the Left and Right MFD's have a very simple relationship:

$$H(z) = B(z)\,[a(z)]^{-1} = [a(z)\,\mathrm{I}_M\,]^{-1}\ \hat{B}(z) = \hat{A}(z)^{-1}\ \hat{B}(z)\ , \tag{26}$$

where I_M is the $M \times M$ unity matrix. The ARMA model fitting procedure gives direct estimates of the Left MFD in this case, since

$$\begin{aligned} y(k) &= -\sum_{i=1}^{n} [a_i\,\mathrm{I}_M]\,y(k-i) + \sum_{i=1}^{n} \hat{B}_i\,u(k-i) \\ &= -\sum_{i=1}^{n} a_i\,y(k-i) + \sum_{i=1}^{n} B_i\,u(k-i)\ . \end{aligned} \tag{27}$$

In other words, $B(z) = \hat{B}(z)$. $\hat{A}(z) = a(z)\,\mathrm{I}_M$ and $A(z) = a(z)$.

In the previous discussions we assumed a high SNR situation where the measurement noise was ignored. It is important to note that the measurement noise can be embedded in an ARMA model with a higher dimensional output vector, i.e.

$$y(k) = -\sum_{i=1}^{n} \hat{A}_i\,y(k-i) + \sum_{i=1}^{n} \hat{B}_i\,u(k-i) + e(k)\ ,$$

where

$$e(k) = n(k) + \sum_{i=1}^{n} \hat{A}_i\,n(k-i)\ ,$$

can be rewritten as

$$y(k) = -\sum_{i=1}^{n} \hat{A}_i\,y(k-i) + \sum_{i=0}^{n} \underbrace{[\ \hat{B}_i,\ \hat{A}_i\ ]}_{\hat{C}_i} \underbrace{\begin{bmatrix} u(k-i) \\ n(k-i) \end{bmatrix}}_{w(k-i)\ ,}$$

let $\qquad\qquad\qquad\qquad$ (28)

where

$$\hat{B}_0 = 0, \quad \hat{A}_0 = I \quad , \hat{C}_0 = [\ 0,\ I\ ].$$

The resulting identification problem is that of an ARMA model with a white driving process $w(k)$ and no measurement noise term. An added complication is the special structure of the matrix $\hat{C}_i$, i.e. the identification algorithm will have to guarantee that the last M columns of $\hat{C}_i$ will be identical to the columns of $\hat{A}_i$. This requires some modifications of the identification algorithms presented in the appendices.

It should be noted that the more classical approach of beamforming is naturally related to the right MFD representation, while the identification approach is related to the left MFD. In the single source case, beamforming consists of forming an inner product of the data vector $Y(z)$ with a "steering" vector $V(z)$, to get a scalar signal $S(z)$ which can then be processed further (e.g. to get optimal detection),

$$S(z) = V(z)^T\ Y(z) = V(z)^T\ B(z)\ X(z) \quad .$$

The steering vector is chosen so that it "removes" the numerator polynomial which represents the propagation model, i.e. we choose $V(z)^T$ so that

$$V(z)^T \quad B(z) = z^{-\Delta}$$

and then the received signal $S(z)$ is just a delayed (with some delay Δ) version of the source signal $X(z)$. To see this more clearly, consider the $B(z)$ given in equation (6b). In that case

$$V(z) = \frac{1}{2c} \begin{bmatrix} cz^{-D_2} \\ z^{-D_1} \end{bmatrix}$$

In the multiple-source case $V(z)$ will be a matrix which is chosen so that

$$V(z)^T \; B(z) = diag \; \{ \; z^{-\Delta_i} \; \} \quad . \tag{29}$$

The numerator polynomial $B(z)$ associated with the steering vector $V(z)$, is that of the right MFD. Thus, the operation of beamforming can be thought of as finding the "inverse" of $B(z)$, in the generalized sense of equation (29).

Determining the Number of Sources

In order to perform system identification, we first have to determine the dimensions of the system, i.e. the number of inputs and outputs. The number of outputs is, of course, the number of sensors M and is known. However, the number of inputs is equal to the number of sources N, which may be unknown, or known only imprecisely. Therefore, it will be necessary to repeat the identification process assuming different numbers of inputs. By observing the resulting mean square errors (i.e. model fitting errors) the number of sources N can be estimated. Typically, there will be a sharp drop in the error power (or other related measures) when the system dimension exceeds N.

The following procedure is, therefore, suggested:

(i) Get an initial estimate of the number of sources. Such an estimate can be obtained from some preprocessing of the sensor data and by keeping track of the number of sources entering and leaving the area of interest.

(ii) Choose the system input dimension to be larger than the estimated number of sources (i.e. overestimate).

(iii) Perform system identification and observe the error.

(iv) Decrease the input dimension by 1 and repeat (iii).

(v) Stop when a sharp increase in error is observed. Choose the previous dimension to be the number of sources. Use the corresponding system parameters $(A_N(z),\ B_N(z))$ to estimate source location and "signature".

If good initial estimates of the number of sources is available, the amount of additional computation will be minimized. It is also possible to derive algorithms that are recursive on the assumed number of sources, such that computational savings can be made. Other schemes for tackling this problem are possible, especially in the case of moving sources observed over a sufficiently long period of time. Details will be presented in future papers.

IV. Some Extensions

The system identification approach can be applied to a wider class of problems than those discussed so far. In particular, it need not be restricted to AR sources and MA propagation models. In this section we present several interesting extensions of these problems and indicate how the proposed approach can be used to solve them.

IV.1. ARMA Sources

Many sources of interest can be adequately modeled as the output of an AR model driven by white noise. (In fact any source with rational spectral density function can be represented this way, provided that the order of the model is sufficiently high). However, it is sometimes necessary to consider ARMA models rather than AR models. To see the kinds of difficulties that arise, consider multiple-source vector x , where

$$X(z) = B(z)\, A_s^{-1}(z)\, V(z) \tag{30a}$$

and

$$A_s(z) = diag\ \{\ a_i(z)\ \}\ , \tag{30b}$$

$$B_s(z) = diag\ \{\ b_i(z)\ \}\ . \tag{30c}$$

The subscript s indicates elements of the transfer function associated with the source models. As we have seen before, the propagation model is given by a polynomial matrix representing the delays, attenuations and possible multipath propagation for the sources to the sensor. We shall denote this matrix by $B_p(z)$. The sensor output vector is

$$Y(z) = B_p(z)\,X(z) + N(z) = B_p(z)\,B_s(z)\,A_s^{-1}(z)\,V(z) + N(z)$$

$$B_p(z)\;B_s(z) = B(z)$$

Note that the numerator matrix $B(z)$ is a product of the propagation model (B_p) and the source MA part. In order to recover the TDOA information it is necessary to factor B into its components B_p, B_s. This is possible assuming a diagonal structure for $B_s(z)$, i.e.

$$B_s(z) = diag\,\{\, b_{s,i}(z) \,\}\ .$$

Because of this structure, all the elements in a column of $B_s(z)$ have a common factor. In fact $b_{s,i}(z)$ will be a common factor of the elements of the i-th column of $B(z)$. Therefore, the following procedure could be used to perform TDOA estimation:

(i) Estimate $A(z)$, $B(z)$ from the available data

(ii) For each column of $B(z)$ find the common factors. This could be done by computing the roots of the polynomials and looking for "root clusters". The columns of $B(z)$ with the common factors taken out will form $B_p(z)$.

(iii) Estimate TDOA from $B_p(z)$ as in the AR case.

Note, however, that the TDOA information is embodied in differences of polynomial degrees in a given column. Since differences in degrees are unchanged by multiplication of all the elements with the same polynomial, it may be possible to extract the desired information directly from $B(z)$. This is easy to see for the case of no multipath. To illustrate this consider

$$B_p(z) = \begin{bmatrix} z^{-D_1} & * \\ cz^{-D_2} & * \end{bmatrix} \quad , \quad B_s(z) = \begin{bmatrix} b_{s,1}(z) & 0 \\ 0 & b_{s,2}(z) \end{bmatrix} \quad , \tag{31a}$$

where * are entries of no interest to this discussion, and with

$$b_{s,1}(z) = g_0 + g_1 z^{-1} + \ . \ . \ . + g_m z^{-m} \quad . \tag{31b}$$

Then the first column of $B(z) = B_p(z)\, B_s(z)$ will be

$$\begin{bmatrix} z^{-D_1}\, b_{s,1}(z) \\ cz^{-D_2}\, b_{s,1}(z) \end{bmatrix} \begin{matrix} = \\ = \end{matrix} \begin{bmatrix} g_0 z^{-D_1} + g_1 z^{-D_1-1} + \ . \ . \ . + g_m z^{-D_1-m} \\ cg_0 z^{-D_2} + cg_1 z^{-D_2-1} + \ . \ . \ . + cg_m z^{-D_2-m} \end{bmatrix} \quad . \tag{32}$$

Note that if we look at the first nonzero terms of these polynomials ($g_0 z^{-D_1}$, $cg_0 z^{-D_2}$) and take the difference of their degrees, we get the correct TDOA estimate ($D_2 - D_1$).

In summary: having an ARMA source model, rather than a pure AR model does not significantly complicate the TDOA estimation problem.

IV.2. ARMA Propagation Models

The simplest form of multipath propagation is caused by a signal being reflected by some object which lies outside the direct line of propagation, causing an extraneous delayed version of the signal to arrive at the sensors. This type of multipath propagation can be represented by a MA model of the type used in earlier sections ($B_p(z)$). Often, a more complicated type of propagation occurs. The signal can undergo multiple reflections of the type encountered, for instance, when a sound wave propagates in a room (e.g., the "cocktail party" problem: locating people who are speaking inside a room). The sound is reflected from one wall, bounces off the opposite wall and again from the first wall. Similar effects occur when seismic waves propagate in the earth or when sound propagates in the ocean (reflected from the water/air and water/ocean floor interfaces). This phenomena is best modeled by an AR type model.

In general, a realistic propagation model will combine both types of multipath propagation and will therefore be an ARMA model, i.e.

$$Y(z) = B_p(z)\, A_p^{-1}(z)\, X(z) + N(z) \quad . \tag{33}$$

The structure of the propagation model depends, of course, on the particular application being considered and on the physical properties of the propagation medium. In its simplest form $A_p(z)$ will have a diagonal form, meaning that the propagation from a given source to all sensors has the same kind of propagation effects, except for different delays which are represented in $B_p(z)$. In general, more complicated forms of propagation are possible in which case $A_p(z)$ will be non-diagonal.

Estimating the source location and spectrum requires estimating the system transfer function $B(z)\, A^{-1}(z)$ and then "factoring out" $B_p(z)$ and $B_s(z)\, A_s^{-1}(z)$. We have seen how this can be done for MA propagation models. The situation is more complicated in the ARMA case, but under certain assumptions it can still be done. To see this consider first the simpler case where the source model is purely autoregressive, and we also assume that $A_s(z)$, $A_p(z)$ are diagonal,

$$H(z) = B_p(z)\, A_p^{-1}(z)\, A_s^{-1}(z) = B_p(z)\, [\, A_s(z)\, A_p(z)\,]^{-1} \quad . \tag{34}$$

The TDOA estimates can be found directly from $B(z) = B_p(z)$, as before. The factoring of $A(z)$ into $A_s(z)$ and $A_p(z)$ is not possible in general. However, in many cases of interest it may still be performed. Consider the situation where the time dependence of the source spectrum and the propagation model are different. For example, consider a nonstationary source, like speech, which stays fixed in space so that the propagation model is time-invariant. By repeating the computation of $A(z)$ over several time intervals and computing its roots, $A_p(z)$ will be found from those roots which do not change, while $A_s(z)$ will correspond to the roots which are varying from one time interval to another. As another example, consider a statistically stationary source, which is moving, causing the propagation model to change. Here $A_s(z)$ will correspond to the fixed roots of $A(z)$ and $A_p(z)$ to the changing roots.

If either $A_s(z)$ or $A_p(z)$ are non-diagonal, the nonuniqueness of the MFD representation will cause some difficulty in getting the TDOA information from $B(z)$. The question of getting the proper "canonical" form of $B(z)$, $A(z)$ for a general nondiagonal $A(z)$ is not completely resolved at this time.

When both the source model and the propagation model are of the ARMA type, the system transfer function is given by

$$H(z) = B_p(z)\, A_p^{-1}(z)\, B_s(z)\, A_s^{-1}(z) \ . \tag{35}$$

In this case the relationship between $A(z)$, $B(z)$ and $A_p(z)$, $A_s(z)$, $B_p(z)$, $B_s(z)$ is not as straightforward as before. Under certain assumptions, the source and propagation models can still be factored. We will not, however, discuss this case here.

IV.3. Direct Estimation of Source Location

The set of TDOA's contains all the information required to locate the sources, i.e. source bearing and range or source coordinates. Thus, estimating the source location can be performed in two steps: First estimate the TDOA's (Δ_{ij}) using the method described in the previous section. Then compute the source location from the set of Δ_{ij}'s using the approach presented in [1,2]. The relationship between the source coordinates (or equivalently: its range and bearing) and the TDOA is given by the problem geometry. For each pair of sensors in 2-D plane, a given TDOA corresponds to a source located on a hyperbolic line of position. Different pairs of sensors and their TDOA define different hyperbolic lines of position. The source has to lie on the intersection of these various lines. These geometric facts can be translated into algorithms for computing source location, as outlined in [1,2].

A different approach would be to try and estimate directly the source coordinates. Since the functional relationship between the source coordinates and the TDOA's is known, it is possible to re-formulate the problem as an estimation problem for those coordinates. The resulting equations are, however, quite complex and possible solutions, especially the ones that lend themselves to distributed computing, are still under investigation.

IV.4. Moving Sources

When the sources and the sensors are fixed in space the propagation model will usually be time-invariant. (This is not necessarily true, since the propagation media may be nonstationary, but for the moment we ignore this case). Thus, the corresponding ARMA coefficients will be constant. If there is source motion, the ARMA coefficients will be time-varying. Provided that these changes are not fast compared to the system sampling rate, the identification algorithms will be capable of tracking the system parameter changes and thus the source location.

Another approach to the moving source problem is to assume a priori a specific form for the time variation of the model and thus reduce the problem again to estimating constant coefficients. For example, the MA coefficients of the propagation model may turn out to have a polynomial form

$$B_i(t) = B_{i,0} + B_{i,1}t + B_{i,2}\,t^2 \ . \tag{36}$$

The coefficients $B_{i,0}$, $B_{i,1}$, $B_{i,2}$ of this time polynomial can be related to the source location at some reference time, its velocity and acceleration. Thus estimating the time-varying ARMA coefficients will provide direct estimates of moving source parameters. The proper form of $A_i(t)$, $B_i(t)$ and the resulting algorithms are currently under investigation.

In many applications, the signals received from moving sources undergo significant doppler shifts. These frequency shifts provide information about target velocity and about its location with respect to the sensors. Source location can be estimated by looking at the differences in frequencies of arrival (FDOA) of the signals. The estimation of target location from the set of FDOA's is analogous to the estimation from TDOA's which was described earlier.

The estimation of FDOA's fits nicely into the system identification framework developed in earlier sections. To see this consider a single AR source of the type

$$X(z) = \frac{1}{a(z)} U(z) \quad . \tag{37}$$

The signals received from this source are given by

$$Y(z) = H(z)\,U(z) + N(z) , \tag{38a}$$

with

$$H(z) = \begin{bmatrix} b_1(z)/a_1(z) \\ : \quad : \\ b_M(z)/a_M(z) \end{bmatrix} \tag{38b}$$

and

$$a_i(z) = a(z-z_i) \qquad , \tag{38c}$$

where z_i corresponds to the doppler shift from the source to sensor i. Estimates of the coefficients of $a_i(z)$ can be obtained by using identification techniques of the type described earlier. The doppler shifts (i.e. FDOA's) can then be found from these coefficients as indicated by equation (38c).

In the multisource case $U(z)$ will be a vector and $H(z)$ will have the form of a matrix with rational polynomial entries

$$H(z) = [\ b_{ij}(z) \,/\, a_{ij}(z)\ ] \qquad 1 \le i \le M, \quad 1 \le j \le N. \tag{39}$$

The coefficients of the denominator polynomials $a_{ij}(z)$ contain the FDOA estimates, while the coefficients of numerator polynomials include the TDOA etimates. The details required to substantiate these statements will be described elsewhere, but the basic ideas can be seen from the single AR source case. Note that if the doppler shifts are negligible, the system transfer function $H(z)$ reduces to the form which was presented earlier; the columns of $H(z)$ have a common denominator polynomial, i.e.

$$a_{ij}(z) = a_j(z) \tag{40}$$

and therefore,

$$H(z) = B(z)\, A^{-1}(z) \quad , \qquad A(z) = diag\,\{\, a_j(z) \,\} \ . \tag{41}$$

This equation is identical to equation (18) which was derived while ignoring doppler shifts.

V. Conclusions

In the previous sections we presented a linear systems framework for treating the source location and spectral estimation problem. We believe that the proposed approach has potential advantages over currently used techniques, particularly in situations involving multiple sources and multipath propagation. Application of this approach to real data involves a myriad of practical issues which were not addressed in this paper. These include:

- Optimal detection and TDOA estimation schemes based on the coefficients of $B_p(z)$. Simple thresholding and interpolation (to refine the bearing resolution) are probably suboptimal.
- Probability of detection (P_d) and probability of false alarm (P_{fa}) curves parametrized by signal-to-noise ratios (SNR) for the proposed algorithms.
- Bounds for the bearing resolution achieveable by this approach.
- Validation of the proposed approach by computer simulation.

These and other practical issues will be addressed in future papers, including the presentation of the computer simulations in progress.

The source location problem also served us as a prototype for motivating research into several system-theoretic areas, including:

- Development of efficient, numerically stable algorithms for going from Right-MFD to Left-MFD and vice versa.
- Algorithms for direct estimation of Left-MFD's.

- Uniqueness and identifiability of multi-input multi-output MFD's.
- Identification of ARMA models with time varying parameters.
- Identification of ARMA models with special structure (e.g. the low signal to noise case in section III).
- Identification of systems with certain types of nonlinearities (e.g. those appearing in the direct estimation of the source location, see section IV.3).

These topics are currently under investigation and some partial results were obtained. The details will be presented elsewhere.

The approach outlined in this paper makes it possible to apply a whole range of techniques developed for system identification to the source location problem. Improved performance, the possibility of handling multipath and simultaneous estimation of TDOA and source spectrum are expected features of this approach. In addition, we have seen that the system identification framework leads to a number of interesting system theoretic questions.

References

[1] Schmidt, R. O., "A New Approach to Geometry of Range Difference Location", IEEE Trans. Aerospace and Electronic Systems, Vol. AES-8, No. 6, Nov. 1872, pp. 821-835.

[2] Delosme, J. M., M. Morf and B. Friedlander, "A Linear Equation Approach to Locating Sources from Time-Difference-of-Arrival Measurements," submitted for publication.

[3] Hahn, W. R. and S. A. Tretter, "Optimum Processing for Delay-Vector Estimation in Passive Signal Arrays", IEEE Trans. Information Theory, Vol. IT-19, No. 5, Sept. 1973, pp. 608-614.

[4] Knapp, C. H. and G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay", IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-26, No. 6, Aug. 1976, pp. 320-327.

[5] MacDonald, V. H. and P. M. Schultheiss, "Optimum Passive Bearing Estimation in a Spatially Incoherent Noise Environment", J. Acoustic Soc. Amer., Vol. 46, July 1969, pp. 37-43.

[6] Van Trees, H. L., *Detection, Estimation and Modulation Theory, Pt. I*, Wiley, New York, 1968.

[7] Rockmore, A. J., Lecture Notes on Array Processing, Stanford University, Summer 1977.

[8] Morf, M., B. Friedlander and J. Newkirk, "Tutorial Survey of Algorithms for Locating and Identifying Spatially Distributed Sources and Receivers," Proc. Distributed Sensor Nets Workshop, Carnegie-Mellon University, Pittsburgh, Penn., Dec. 7-8, 1978, pp. 97-108.

[9] Astrom, K. J. and P. Eykhoff, "System Identification - A Survey", Automatica, Vol. 7, Jan. 1971, pp. 123-162.

[10] Eykhoff, P., *System Identification, Parameter and State Estimation*, Wiley, New York, 1974.

[11] Morf, M., T. Kailath and L. Ljung, "Fast Algorithms for Recursive Identification," Proc. Conf. on Decision and Control, Florida, Dec. 1976, pp. 916-921.

[12] Morf, M., "Fast Algorithms for Multivariable Systems", Ph. D. Dissertation, Stanford University, Stanford, California, Aug. 1974.

[13] Morf, M., D. T. Lee, J. R. Nickolls and A. Vieira, "A Classification of Algorithms for ARMA Models and Ladder Realizations," Conf. Rec., 1977 IEEE Int. Conf. on Acoustics, Speech & Signal Processing, Hartford, 1977, pp. 13 - 19.

[14] Soderstrom, I., L. Ljung and I. Gustavsson, "A Comparative Study of Recursive Identification Methods," Report 7427, Dept. Automatic Control, Lund Institute of Technology, Lund, Sweden, 1974.

[15] Goodwin, G. C. and R. L. Payne, *Dynamic System Identification*, Academic Press, New York, 1977.

[16] Morf, M., "Ladder Forms in Estimation and System Identification," Proc. 11th Annual Asilomar Conf. on Circuits, Systems and Computers, Monterey, California, Nov. 7-9, 1977, 424-429.

[17] Morf, M. and D. T. Lee, "Recursive Least Squares Ladder Forms for Fast Parameter Tracking," Proc. 1979 IEEE Conference on Decision and Control, San Diego, California, January 10-12, 1979, pp. 1362-1367.

[18] Burg, J. P.,"Maximum Entropy Analysis," Ph.D. Dissertation,Stanford University, Stanford, California, 1975.

[19] Morf, M, A. Vieira, D. T. L. Lee and T. Kailath, "Recursive Multichannel Maximum Entropy Spectral Estimation," IEEE Trans. Geosci. and Electronics.,

Vol. GE-16, No. 2, April 1978, pp. 85-94.

[20] Friedlander, B., M. Morf, T. Kailath and L. Ljung, "New Inversion Formulas for Matrices Classified in Terms of Their Distance from Toeplitz Matrices", Linear Algebra and Its Applications, J. Linear Algebra and Its Applications, 1979.

[21] Friedlander, B., M. Morf, T. Kailath and L. Ljung, "Extended Levinson and Chandrasekhar Equations for General Discrete-Time Linear Estimation Problems," IEEE Trans. Auto. Cntrl., V. AC-23, 1978, pp. 643-659.

[22] Dickinson, B. W., T. Kailath and M. Morf, "Canonical Matrix Fraction and State-Space Descriptions for Deterministic and Stochastic Linear Systems", IEEE Trans. Automatic Control, Vol. AC-19, No. 6, Dec. 1974, pp. 656-667.

[23] Dickinson, B. W., "Properties and Applications on Matrix Fraction Description of Linear Systems", Ph.D. dissertation, Stanford University, Stanford, California 1974.

Appendix A:

Example of Recursive Identification Algorithms

Perhaps the most basic and most commonly used identification scheme is the least squares (LS) method. This method is archetypical for several others, and in particular for those to be considered here. For a thorough treatment of this subject see, e.g. [9], [10], [14], [15].

Problem Statement

Given measured input output data we are interested in fitting an ARMA model to this data, as discussed in section II. Recall that the ARMA model parameters are given by the matrices A_k *and* B_k , hence they are to be determined from recorded input-output data $(y(t),\ u(t),\ t = 1, \ . \ . \ . \ . \ , N)$. *The* A *parameters represent the* autoregressive part, whereas the B parameters are the moving-average part.

The LS- (Least-Squares) Method amounts to choosing the parameters A_k, B_k so that the "equation error" $e(t)$ (see e.g. equation 7 in section II) is minimal

$$min_{A_k,B_k} \sum_{t=1}^{N} \| e(t) \|^2 =$$

$$min_{A_k,B_k} \sum_{t=1}^{N} \| y(t) + A_1 y(t-1) + \ . \ . \ . \ + A_r y(t-r) - B_1 u(t-1) - \ . \ . \ . \ - B_r \, u(t-r) \|^2$$

We introduce the following notation, denote transpose by ^{T} and define $s = p + m$ and $q - rs$, then the input output data is given by the vector of the measurements

$$Y[t-1, t-r]^T = [\, y(t-1)^T,\ u(t-1)^T, \ . \ . \ . \ , y(t-r)^T,\ u(t-r)^T \,]$$

(a q by 1 vector) .

And the model parameter vector is given by

$$A[1, r]^T = [\, -A_1^T,\ B_1^T, \ . \ . \ . \ . \ , -A_r^T,\ B_r^T \ ]$$

(a p by q - matrix) .

Then the output equations can be written in vector form as

$$y(t) = A[1,r]^{T} \; Y[t-1,t-r] + e(t)$$

and we wish to minimize the equation error by minimizing

$$min_{A} \sum_{t=1}^{N} \| y(t) - A[1,r]^{T} \; Y[t-1,t-r] \|^{2} \quad .$$

Since (2) is a quadratic function in A, in order to find the unknown parameters it is easy to give the minimizing A at time N, denoted by A_N, explicitly via the so-called normal equations

$$R_N \; [\, I, \; -A_N^{T} \;]^{T} = [\; R^{e}_{N}, \; O, \; . \; . \; . \; . \; , \; O \;]^{T}$$

$$\text{or} \qquad R_{N-} \; A_N = R_N .$$

If the inverse of R_N or R_{N-} exists, we can obtain an equation for the estimates of the parameter vector that minimizes the equation error (2)

$$A_N = [\; R_{N-} \;]^{-1} \; R_N,$$

where with $q = r(p+m)$ and $q + s = (r+1)(p+m)$

$$R_N = \sum_{t=1}^{N} Y[t,t-r] \; Y[t,t-r]^{T} \quad (\; q + s \text{ by } q + s \; - \; matrix \;)$$

$$R_{N-} = \sum_{t=1}^{N} Y[t-1,t-r] \; Y[t-1,t-r]^{T} \quad (\; q \text{ by } q \; - \; matrix \;)$$

are the joint sample covariances of the measured input output data, and

$$R_N = \sum_{t=1}^{N} Y[t-1,t-r] \; y(t)^{T} \quad (\; q \text{ by } p \; - \; matrix \;)$$

is the sample cross-covariance of the "current" output $y(t)$ and the past inputs and outputs.

$$R^{e}_{N} = \sum_{t=1}^{N} e(t) \; e(t)^{T} \quad (\; q \text{ by } p - matrix \;)$$

R^{e}_{N} is the sample prediction error covariance of the output $y(t)$.

As is very well known, e.g. [9,10], this solution can be written recursively in time t, i.e. as a function of the in real-time measured input and output data $y(t)$ and $u(t)$.

$$A_{t+1} = A_t + k^{e}_{t+1} \; e_{t+1}^{T}$$

The parameter estimates are updated by increment that is proportional to $e(t+1)$, the error in predicting the next output $y(t+1)$.

$$
\begin{aligned}
e_{t+1}{}^{T} &= y_{t+1}{}^{T} - y[t, t+1-r]^{T} A_t \\
&= y[t+1, t+1-r]^{T} \; [\; I, -A_t{}^{T} \;]^{T}
\end{aligned}
$$

The gain k^e is obtained similarly via the prediction error covariance P_t of the parameters.

$$
\begin{aligned}
k^{e}{}_{t+1} &= P_{t+1}{}^{T} \; y[t, t+1-r] = k_{t+1} \; r^{e}{}_{t+1}{}^{-1} \\
k_{t+1} &= P_t \; y[t, t+1-r]
\end{aligned}
$$

prediction error covariance $r^{e}{}_{t+1}$ of the data y_{t+1}

$$
\begin{aligned}
r^{e}{}_{t+1} &= 1 + y[t, t+1-r]^{T} \; P_t \; y[t, t+1-r] \\
&= 1 + y[t, t+1-r]^{T} \; k_{t+1} \\
P_{t+1} &= P_t - k_{t+1} \; r^{e}{}_{t+1}{}^{-1} \; k_{t+1}{}^{T}
\end{aligned}
$$

where the recursions for P_t are recursions for the inverse of R, the sample covariance of the input output data, since

$$
P_t = [\; R_{t-} \;]^{-1}
$$

The above set of equations can be regarded as a state estimator or Kalman filter, where the state is a vector of the constant parameters in A that are to be identified or estimated. The output equation of the model is given by equation 7 in section II, and the state equation is trivial since the state is a constant vector. For vector measurements ($p > 1$) the equations above differ actually from the usual state estimator equations since then the "state" is a p by q matrix instead of a pq by 1 vector, which has certain implications on the dimensions of P, (here a q by q and not a pq by pq matrix); however it is not very hard to recast this convenient form into the commonly used state-space format, hence using all the available techniques for analysing and implementing such algorithms.

Appendix B:

Pole-Zero or Autoregressive Moving-Average(ARMA) Forms

In this appendix we discuss fast algorithms for pole-zero or autoregressive moving-average (ARMA) modeling. These forms have many advantages over the forms presented in Appendix A, see example [13], [16], [17]. They appear especially well suited for real-time applications using fast pipeline type processors.

B.1 Joint Innovations Representation of the ARMA Process

Given a pole-zero (ARMA) model of the type discussed in section II and in Appendix A its equations can be rewritten as

$$y_t + A_1 y_{t-1} + \dots + A_N y_{t-N} - B_1 u_{t-1} - \dots - B_N u_{t-N} = B_0 u_t ,$$

and in matrix notation, we have

$$\mathbf{a}_N^{\mathsf{T}} \; \mathbf{y}_t \; - \; \mathbf{b}^*{}_N^{\mathsf{T}} \; \mathbf{u}_t \; = \; B_0 u_t ,$$

with

$$\begin{aligned} \mathbf{a}_N^{\mathsf{T}} &= [\, I_m , \; A_1, \; A_2, \; \dots , \; A_N \,], \\ \mathbf{b}^*{}_N^{\mathsf{T}} &= [\, O_m , \; B_1, \; B_2, \; \dots , \; B_N \,], \\ \mathbf{y}_t^{\mathsf{T}} &= [\, y_t^{\mathsf{T}} , \; \dots , \; y_{t-n}^{\mathsf{T}} \,], \\ \mathbf{u}_t^{\mathsf{T}} &= [\, u_t^{\mathsf{T}} , \; \dots , \; u_{t-n}^{\mathsf{T}} \,], \end{aligned}$$

leading to the following augmented equation

$$\begin{bmatrix} \mathbf{a}_N^{\mathsf{T}} & -\mathbf{b}^*{}_N^{\mathsf{T}} \\ O & \delta_0^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{y}_t \\ \mathbf{u}_t \end{bmatrix} = \begin{bmatrix} B_0 u_t \\ u_t \end{bmatrix} ,$$

where δ_0 is the first m *by* m block unit vector.

This embedded model can be interpreted as a 2-channel AR model of the joint process $\{ y_t , \; u_t \}$. Here, the right hand side of the augmented equation is equal to the joint innovations or prediction errors of $\{ y_t, u_t \}$, since

$$\epsilon_t = \begin{bmatrix} \epsilon^y{}_t \\ \\ \epsilon^x{}_t \end{bmatrix} = \begin{bmatrix} y_t - \hat{y}_{t|t-1} \\ \\ u_t - \hat{u}_{t|t-1} \end{bmatrix} = \begin{bmatrix} B_0 u_t \\ \\ u_t \end{bmatrix} .$$

Indeed if we apply a simple interleaving permutation of (1, 3, 5, . . . , $2N+1$, 2, 4, 6, . . . $2N+2$) to the augmented equation, we have a 2-channel one-step predictor of the autoregressive form

$$A_N{}^{\mathsf{T}} z_t = \epsilon_t ,$$

where

$$A_N = \left[\begin{bmatrix} I_m & O_m \\ \\ O_m & I_m \end{bmatrix} , \begin{bmatrix} A_1 & -B_1 \\ \\ O_m & O_m \end{bmatrix} , \ldots , \begin{bmatrix} A_N & -B_N \\ \\ O_m & O_m \end{bmatrix} \right] ,$$

$$z_t{}^{\mathsf{T}} = [y_t{}^{\mathsf{T}}, \; u_t{}^{\mathsf{T}}, \; . \; . \; . \; . \; y_{t-N}{}^{\mathsf{T}}, \; u_{t-N}{}^{\mathsf{T}}] .$$

If the covariance of the input/output data is given
the problem of finding the linear least-squares predictor
is reduced to solving a normal equation of the following form

$$R^{yu}{}_N \, A_N = [\, R^{\epsilon}{}_N , \; 0, \; 0, \; . \; . \; . \; 0 \,]^{\mathsf{T}} ,$$

where the joint covariance is given by (E is the expection operator)

$$R^{yu}{}_N = E \, z_t \, z_t{}^{\mathsf{T}} = \begin{bmatrix} R^{yu}{}_0 & R^{yu}{}_1 & . & . & . & R^{yu}{}_N \\ R^{yu}{}_1{}^{\mathsf{T}} & R^{yu}{}_0 & . & . & . & . \\ . & . & . & . & . & . \\ \\ . & . & . & . & . & . \\ R^{yu}{}_N{}^{\mathsf{T}} & & . & . & . & R^{yu}{}_0 \end{bmatrix} ,$$

$$R^{yu}{}_n = E \begin{bmatrix} y_t \\ u_t \end{bmatrix} [\, y_{t-n}{}^{\mathsf{T}} \; u_{t-n}{}^{\mathsf{T}} \,] ,$$

and the prediction error or innovation covariance is defined as

$$R^{\epsilon}{}_N = E \, \epsilon_t \, \epsilon_t{}^{\mathsf{T}} .$$

$$R^{yu}{}_0 = \begin{bmatrix} R_y(0) & R_{yu}(0) \\ & \\ R_{yu}(0)^{\mathsf{T}} & I_m \end{bmatrix}, \quad R^{yu}{}_n = \begin{bmatrix} R_y(n) & R_{yu}(n) \\ & \\ O_m & O_m \end{bmatrix}.$$

Since $R^{yu}{}_N$ has a special structure, i.e. a block Toeplitz matrix with blocks of size $2m$ *by* $2m$, the solution to this normal equation can be obtained via fast algorithms, see e.g. [13]. These algorithms again are adapted to efficient computer implementations.

B.2 Least-Squares Recursions for ARMA Models

The variables of interest are in this case the joint forward and backward prediction errors of order n at time T is defined as

$$\epsilon_{n,T} = \begin{bmatrix} \epsilon^y{}_{n,T} \\ \\ \epsilon^u{}_{n,T} \end{bmatrix} = \begin{bmatrix} y_T - \hat{y}_{T|T-1,\ldots,T-n} \\ \\ u_T - \hat{u}_{T|T-1,\ldots,T-n} \end{bmatrix} = A_{n,T}{}^{\mathsf{T}}\, z[T:T-n] \; ,$$

$$r_{n,T} = \begin{bmatrix} r^y{}_{n,T} \\ \\ r^\mu{}_{n,T} \end{bmatrix} = \begin{bmatrix} y_{T-n} - \hat{y}_{T-n|T-n+1,\ldots,T} \\ \\ u_{T-n} - \hat{u}_{T-n|T-n+1,\ldots,T} \end{bmatrix} = B_{n,T}{}^{\mathsf{T}}\, z[T:T-n] \quad .$$

The covariance matrix for the joint process is thus similar to that given in Section B.1 and taking advantage of its special structure, one can obtain recursions for the 2-channel AR case, i.e.

$$\begin{aligned} A_{n+1,T} &= A_{n,T} - D^{\mathsf{T}}{}_{n+1,T}\, R^{-r}{}_{n,T-1}\, B_{n,T-1} \\ B_{n+1,T} &= B_{n,T-1} - D_{n+1,T}\, R^{-\epsilon}{}_{n+1,T}\, A_{n,T} \; . \end{aligned}$$

In the two-channel case, $D_{n+1,T}$, the partial correlation matrices exhibit additional structures, due to the embedding of the white input process of the ARMA model into an AR model. Since all future inputs are assumed to be uncorrelated with the present and past outputs, half the elements $D_{n+1,T}$ matrix are approximated by zeros. Indeed the partial correlation matrices are equivalent to the ARMA model parameter estimates, see e.g. [19], and they are now given by

$$\Delta_{n+1,T} = \begin{bmatrix} \Delta^{y}{}_{n+1,T} & O_m \\ & \\ \Delta^{u}{}_{n+1,T} & O_m \end{bmatrix} = \begin{bmatrix} \sum_{t=n+1}^{T} y_{t-n-1} \epsilon^{y}{}_{n,T}(t)^{\mathsf{T}} & O_m \\ & \\ \sum_{t=n+1}^{T} u_{t-n+1} \epsilon^{y}{}_{n,T}(t)^{\mathsf{T}} & O_m \end{bmatrix}.$$

The time-updates for these partial correlations are obtained in a similar manner as in the single channel case

$$\Delta^{y}{}_{n+1,T+1} = \Delta^{y}{}_{n+1,T} + [\ \frac{r^{y}{}_{n,T}{}^{\mathsf{T}}\ \epsilon^{y}{}_{n,T+1}{}^{\mathsf{T}}}{1 - \gamma_{n-1,T}}\ ]$$

$$\Delta^{u}{}_{n+1,T+1} = \Delta^{u}{}_{n+1,T} + [\ \frac{r^{u}{}_{n,T}{}^{\mathsf{T}}\ \epsilon^{y}{}_{n,T+1}{}^{\mathsf{T}}}{1 - \gamma_{n-1,T}}\ ]$$

In order to obtain recursions for the ARMA model parameters, inversions of the joint $2m$ *by* $2m$ prediction error covariances, $R^{\epsilon}{}_{n,T}$ and $R^{r}{}_{n,T}$ are needed. These inversions are simplified by first decomposing them into upper-diagonal-lower form and then taking inverses. Examples of the different forms of the fast ARMA model parameter estimation algorithms can be found in [11,12,13,16,17,19]. The numerical properties of these algorithms are still under investigation; however, certain ladder canonical forms that are also described in these references, appear to be most desirable properties for implementations.

Estimating Source Location from Time Difference of Arrival: A Linear Equation Approach

J.M. Delosme*, M. Morf* and B. Friedlander**,

Abstract

A new framework is presented for the problem of estimating source location from a set of time difference of arrival (TDOA) measurements. The proposed approach is based on the fact that three sensors with their set of TDOA's determine a straight line-of-position for the source. It is shown that the source location can be found as the solution to a set of *linear equations*, with the number of equations being equal to the number of sensors. The proposed approach has several desirable features including: the availability of *efficient* recursive algorithns for solving the resulting linear equations, the possibility of *distributed implementation* of these algorithms, and the applicability of *optimal estimation* techniques to this type of equations. Some extensions of this problem formulation to the three-dimensional case, to geopositioning on the surface of a sphere and to source velocity estimation, are also discussed.

* Information Systems Laboratory, Stanford University, Stanford, CA 94305.

** Systems Control Inc., Palo Alto, CA 94304.

This work was supported by the Defense Advanced Research Projects Agency through contract MDA 903-78-C-0179, and the facilities at the Stanford Artificial Intelligence Laboratory.

I. Introduction

A common way of finding the location of a point is by measuring the range differences to several other points whose location is known. In various navigation system (e.g. Loran, Decca, Omega) the measurement consists of observing differences in the time-of-arrival of signals from sources at the known points to a receiver located at the unknown point. The time difference corresponds to a range difference if a constant velocity of propagation is assumed. In the passive sonar case signals travel from a source whose location is to be estimated to sensors with known positions.

The standard approach to estimating source location is based on finding hyperbolic lines of position (LOP). The measurement of a certain TDOA at two sensor locations, determines a hyperbola. Each point on the hyperbola has the same range difference to the two sensors -- see Figure 1. If more than two sensors provide TDOA measurements, several LOP's are generated, one for each pair of sensors. Since the LOP's will, in general, be different and since the source must lie on all of them, their intersection will provide an estimate of the source location.

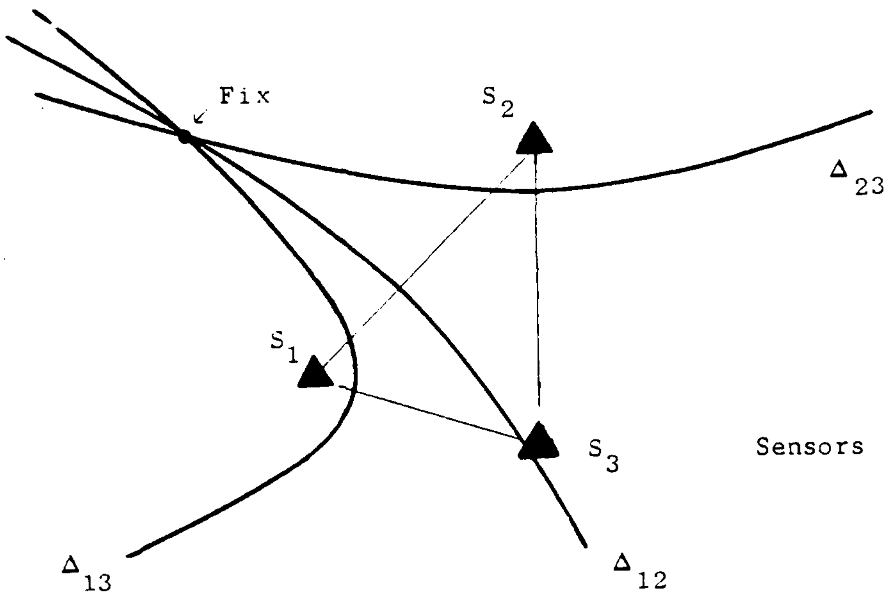


Figure 1 : Hyperbolic Lines of Position

The hyperbolic LOP approach has several serious drawbacks: the analytic computation of the intersection location is very cumbersome; these solutions do not appear to be easily extended to other situations such as calculating source velocity from TDOA-s and their rates of change or computing source location from Doppler measurements. Furthermore, the complexity of the solutions makes error analysis very difficult.

An alternative approach which circumvents many of these difficulties was proposed by Schmidt [1]. His approach is based on the idea that three sensors with their set of TDOA-s determine a *straight* line of position. In fact, that line is the major axis of a general conic which passes through the three sensors -- see Figure 2.

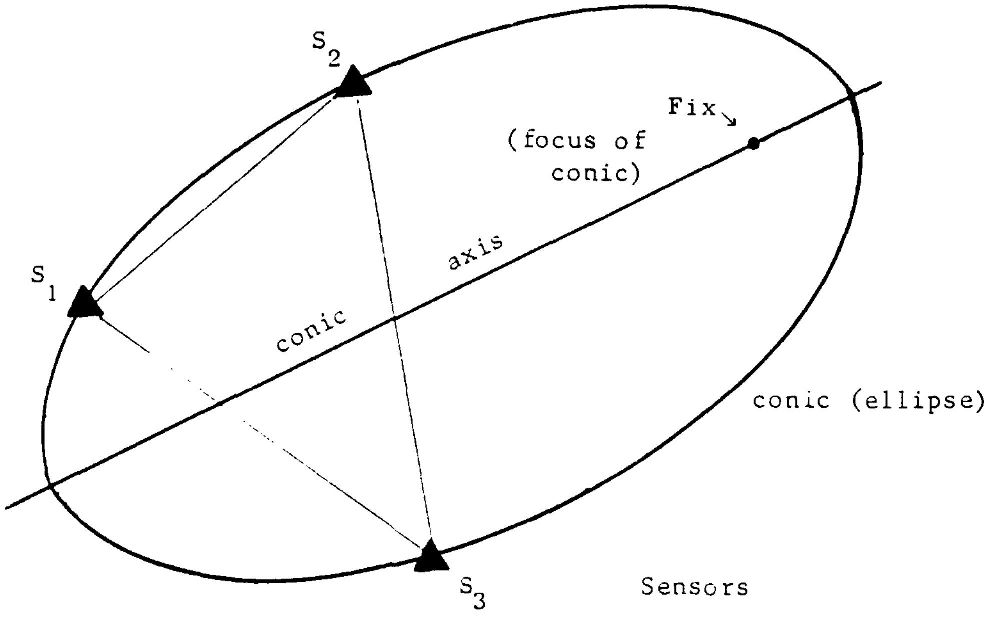


Figure 2 : Location on the Conic Axis

When more than three sensors are available, several straight LOP-s are generated and their intersection provides the estimate of the source location. Schmidt [1] proposes to compute the intersection point from a set of $\binom{N}{3}$* linear equations, N being the number of sensors. This approach, while alleviating some of the difficulties of the hyperbolic LOP approach, still has several drawbacks. The main difficulty lies in the use of redundant information: the formulation leads to a set of $\binom{N}{3}$ equations, while in a sense there are only N independent pieces of information corresponding to the range from each sensor to the source. This redundancy creates some inconsistencies in the case of noisy measurements. The straight LOP approach and the difficulties associated with it are described in section II.

* where $\binom{N}{3}$ is defined as $\frac{N!}{(N-3)!3!}$.

In this paper we propose a different problem formulation which is based on the straight LOP approach. The source location is found by solving a set of *N linear* equations which contain the sensor locations and the source-sensor ranges. This set of equations contains all the relevant sensor data *without* redundancy. We also show how the source-sensor range can be estimated in a consistent optimal way (in the least-squares sense) from the TDOA or TOA measurements. Our problem formulation and the resulting solution algorithms are described in section III.

The proposed formulation of the source location problem has a number of attractive features, which are mostly related to the linearity of these equations:

- Efficient algorithms are available to solve this type of equations

- Algorithms which are recursive in the number of sensors have been derived. Thus, initial estimates can be obtained from a small set of sensors, and these estimates can be updated as information from more sensors becomes available.

- The linearity of the equations makes it possible to implement the computations in a distributed fashion. This is particularly important in the context of distributed sensor nets [2].

- The new formulation is well suited to the development of optimal estimation algorithms.

- The proposed approach lends itself naturally to extensions to several classes of problems, including: source location in three dimensions, geopositioning on the surface of a sphere and estimating target velocity frcm range rate measurements.

These extensions are presented in section IV.

We believe that the approach proposed in this paper provides a proper framework for handling the source location problem. A detailed evaluation of the approach and comparison to currently used techniques is now under way and will be reported later.* We are also in the process of deriving optimal estimation algorithms for the source location problem, as outlined in section III

* A similar approach has been developed and tested on real data and found to perform satisfactorily [18].

II. The Source Location Problem

The solution of the "position fixing" problem using the hyperbolic LOP has been treated quite extensively in the context of navigation systems like Loran [3], Omega [4], and the Global Positioning Systems (GPS) [5]. The complexity of the equations resulting from this approach led to a solution based on various approximation techniques involving linearization and Taylor series expansion [6] - [8]. Since the hyperbolic LOP approach is well documented [9] - [11], we will not describe it here in detail. Instead, we will provide in this section a description and an analysis of the alternative straight LOP approach proposed by Schmidt.

The Straight Line-of-Position Approach

For ease of exposition, attention will be restricted to two-dimensional position location in a plane. Extension of the ideas to more complicated situations will be presented in section IV.

Consider a single source located at a point (x_0, y_0) in the plane and N sensors located at points (x_i, y_i), $i = 1, \ldots, N$, as depicted in Fig. 3. We shall denote by r_i the distance from the i-th sensor to the source, and by Δ_{ji} the range difference for sensors i and j. Then,

$$\Delta_{ji} = r_i - r_j$$

and,

$$r_i^2 = (x_0 - x_i)^2 + (y_0 - y_i)^2 .$$

It is shown in [1] that for any group of three sensors $\{S_i, S_j, S_k\}$, a straight line of position is defined by the relationship,

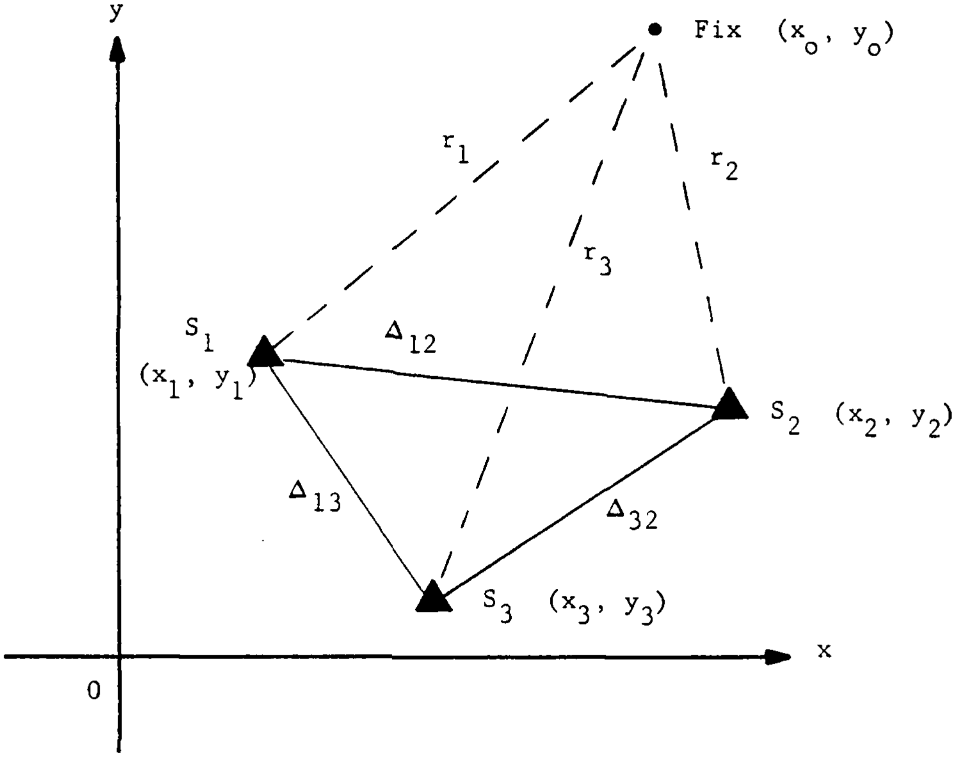


Figure 3: Problem Geometry

$$[\, x_i\,\Delta_{jk} + x_j\,\Delta_{ki} + x_k\,\Delta_{ij}\,]\,x + [\, y_i\,\Delta_{jk} + y_j\,\Delta_{ki} + y_k\,\Delta_{ij}\,]\,y = \frac{1}{2}\,[\,\Delta_{ij}\,\Delta_{jk}\,\Delta_{ki} + a_i^2\,\Delta_{jk} + a_j^2\,\Delta_{ki} + a_k^2\,\Delta_{ij}\,]\ . \qquad (1)$$

where

$$a_i^2 = x_i^2 + y_i^2\ .$$

Schmidt [1] proposes to use the set of linear equations obtained by writing down equation (1) for all $\binom{N}{3}$ possible sensor triads. His solution then consists of solving this set of equations to find the common intersection point (x, y) which in the noiseless case will be exactly the source location (x_0, y_0). Some alternative solution methods are also suggested, based on the observation that the source lies on one of the foci of an ellipse passing through the three sensors. Here, however, we will concentrate on the set of $\binom{N}{3}$ linear equations represented by (1). Our aim is to explain some of the difficulties associated with Scnmidt's proposed solution.

The Noise Free Case

We first note that when the range differences are obtained without errors, Schmidt's set of $\binom{N}{3}$ equations is really equivalent to a related set of only N equations. To see this note that

$$\begin{aligned} \Delta_{ij} \Delta_{jk} \Delta_{ki} &= (r_j - r_i)(r_k - r_j)(r_i - r_k) \\ &= -r_i^2 \Delta_{jk} - r_j^2 \Delta_{ki} - r_k^2 \Delta_{ij} \ . \end{aligned} \tag{2}$$

Therefore, equation (1) may be rewritten as:

$$\begin{aligned} & [x_i \Delta_{jk} + x_j \Delta_{ki} + x_k \Delta_{ij}] x + [y_i \Delta_{jk} + y_j \Delta_{ki} + y_k \Delta_{ij}] y = \\ & \frac{1}{2} [(a_i^2 - r_i^2) \Delta_{jk} + (a_j^2 - r_j^2) \Delta_{ki} + (a_k^2 - r_k^2) \Delta_{ij}] \ . \end{aligned} \tag{3}$$

Let us introduce the linear operator O whose operation on any station dependent quantity u and any triad $\{S_i, S_j, S_k\}$ is defined by:

$$O_{ijk} [u] = det \begin{bmatrix} 1 & r_i & u_i \\ 1 & r_j & u_j \\ 1 & r_k & u_k \end{bmatrix} \ . \tag{4}$$

Using this operator, the original set of equations satisfied by the source coordinates (x_0, y_0) may be rewritten more compactly as:

$$O_{ijk} [x x_0] + O_{ijk} [y y_0] = O_{ijk} [(a^2 - r^2)/2] \ , \tag{5}$$

for all triads ijk.

By linearity of the operator this equation is equivalent to:

$$O_{ijk} [\, x\, x_0 + y\, y_0 - (a^2 - r^2)/2 \,] = 0 \quad \text{for all triads} \quad . \tag{6}$$

Since the last equation holds for every triad, $x\, x_0 + y\, y_0 - (a^2 - r^2)/2$ belongs to the kernel of O, i.e. there exist two constants b and c such that

$$x_i\, x_0 + y_i\, y_0 - (a_i^2 - r_i^2)/2 = c + b\, r_i \quad i = 1, N \quad . \tag{7}$$

Conversely the set of N *linear* equations with four unknowns x_0, y_0, b, c

$$x_i\, x_0 + y_i\, y_0 - r_i\, b - c = (a_i^2 - r_i^2)/2 \quad i = 1, N \tag{8}$$

gives, by application of the operator O, the original set of $\binom{N}{3}$ equations. Therefore the two sets of equations (1) and (8) are *equivalent*.

Redundancy for Noisy Measurements

In reality the range difference measurements are corrupted by noise. To see how this modifies our previous conclusion, more insight into the nature of the noise is needed. In particular, we have to distinguish between two ways in which the TDOA estimates may be obtained:

(i) Differencing Epoch or Travel Times

If the source emits narrow, well defined pulses of energy, the sensor will be able to determine the exact time those pulses were received. Since the sensors do not know the time at which the pulse was transmitted, they are unable to compute the travel time directly. However, by differencing the recorded arrival times of two sensors, the difference in travel times (TDOA) is found, since the unknown "starting time" cancels out.

(ii) Cross Correlation

When the sensor emits a noise-like continuous signal which has no easily recognized features, the TDOA is usually estimated by cross correlating the signals of the two sensors. See, e.g. [12] for correlation techniques and [13] for an ARMA modeling approach.

For both methods the error in the TDOA estimate, or equivalently, in the range difference Δ, may be considered as additive and (approximately) uncorrelated with Δ. While in the first case this noise is the sum of two independent components, one due to each sensor, in the second case such a decomposition does not generally exist.

Redundancy for Case (i)

More precisely for *type* (i) techniques the available measurements have the form $\tilde{r}_i + \nu_i$ where $\tilde{r}_i = \tilde{r}_i + \bar{r}$, and $\bar{r}$ being an unknown quantity identical for all the sensors S_i, wheras ν_i is the additive noise. Therefore the measured range differences are obtained as:

$$\Delta_{ji} = (\tilde{r}_i + \nu_i) - (\tilde{r}_j + \nu_j) \ . \qquad (9)$$

It is easy to check that equations (2) through (8) still hold if r_i is replaced by the measurement $\tilde{r}_i + \nu_i$. Consequently, when Δ_{ji} is defined by (9), the set of N equations

$$x_i x_0 + y_i y_0 - (\tilde{r}_i + \nu_i) b - c = (a_i^2 - (\tilde{r}_i + \nu_i)^2)/2 \ , \quad i = 1, N \qquad (10)$$

is equivalent to the original set of equations.

Hence, if the TDOA's are obtained by differencing of travel times, the original set of equations contains highly *redundant* information and can be replaced by a much smaller set of equations (10).

Inconsistency or Redundancy for Case (ii)

For *type* (ii) techniques the range differences have the form:

$$\Delta_{ji} = [\, r_i - r_j \,] + v_{ji} \ , \tag{11}$$

where the additive noise cannot in general by decomposed as $\nu_i - \nu_j$. Some information about the noise on the Δ's appearing in equation (. is readily available by summing up around the loop *ijk* : $\Delta_{ji} + \Delta_{jk} + \Delta_{ki} = \Sigma$, instead of zero in absence of noise. This information should be used to improve the coefficients of (1). One natural way to do this, called *TDOA averaging* in [1], is to correct the Δ's symmetrically:

$$\begin{aligned} \hat{\Delta}_{ij} &= \Delta_{ij} - \frac{\Sigma}{3} \\ \hat{\Delta}_{jk} &= \Delta_{jk} - \frac{\Sigma}{3} \\ \hat{\Delta}_{ki} &= \Delta_{ki} - \frac{\Sigma}{3} \ , \end{aligned} \tag{12}$$

so that

$$\hat{\Delta}_{ij} + \hat{\Delta}_{jk} + \hat{\Delta}_{ki} = 0 \ ,$$

and use those "corrected" range differences in the computation of the coefficients of the straight LOP (1) . Application of the same procedure to the triad $\{ S_i, S_j, S_l \}$ yields:

$$\hat{\Delta}_{ij} = \Delta_{ij} - \frac{\Sigma'}{3}$$
$$\hat{\Delta}_{jl} = \Delta_{jl} - \frac{\Sigma'}{3}$$
$$\hat{\Delta}_{li} = \Delta_{li} - \frac{\Sigma'}{3} \quad , \qquad (13)$$

where $\Sigma' = \Delta_{ij} + \Delta_{jl} + \Delta_{li}$. Thus two expressions for $\hat{\Delta}_{ij}$, which are in general *inconsistent*, have been obtained.

The fact that the TDOA averaging procedure yields different estimates of the same range difference is linked to its inherent suboptimality; it is possible to find a better estimator of the range differences and, simultaneously, to get rid of any inconsistency by using *all* the information available about the noise on the Δ's to find the $\hat{\Delta}$'s .

Since the exact values of the range differences have the property $\Sigma \Delta = 0$, it is natural to require that $\Sigma \hat{\Delta} = 0$ for *any* closed path or loop (involving three sensors or more). This condition is equivalent to stating the existence of a set of variables $\{\tilde{r}_i, \ i = 1, \ N\}$ such that $\hat{\Delta}_{ji} = \tilde{r}_i - \tilde{r}_j$. Furthermore the inconsistencies will automatically disappear in a procedure which evaluates the $\hat{\Delta}$'s as $\hat{\Delta}_{ji} = \tilde{r}_i - \tilde{r}_j$ where the $\binom{N}{2}$ measurements of Δ are used to estimate the $\tilde{r}$'s. The rule for obtaining the set $\{\tilde{r}_i, \ i = 1, N\}$ which minimizes the variance of the error between the estimated $\hat{\Delta}$'s and the true range differences will be called here "*generalized TDOA averaging*" and will be described in the next section.

The coefficients of the original set of equations should be computed in terms of the consistent set of $\hat{\Delta}$'s which is the best estimate of the set of true range differences. Since $\hat{\Delta}_{ji} = \tilde{r}_i - \tilde{r}_j$, the original $\binom{N}{3}$ equations are equivalent to the N equations:

$$x_i x_0 + y_i y_0 - \tilde{r}_i b - c = (a_i^2 - \tilde{r}_i^2)/2 \quad , \qquad i = 1, N \tag{14}$$

Therefore, the original set of equations contains *redundant* information when the range differences are obtained by *type* (ii) techniques.

These two examples clearly display the importance of using the proper set of equations when only imprecise measurements are available. In the next section we will look more closely at our proposed framework

III. The Linear Equation Approach

As a byproduct of the analysis of the original set of $\binom{N}{3}$ linear equations in x_0, y_0, the source coordinates, a set of N linear equations satisfied by x_0, y_0 and two other variables b and c was obtained:

$$x_i x_0 + y_i y_0 - \tilde{r}_i b - c = (a_i^2 - \tilde{r}_i^2)/2 \quad , \qquad i = 1, N \tag{15}$$

where the $\tilde{r}_i$'s are estimates of the ranges r_i (obtained modulo an additive term $\bar{r}$ since the original data are range differences.) This set of equations can be derived more directly using the fact that the source is located at the intersection of the N circles $(S_i,\ r_i)$:

$$(x_i - x_0)^2 + (y_i - y_0)^2 = r_i^2 = (\bar{r} + \tilde{r}_i)^2 \qquad i = 1, N \ ,$$

or

$$-2 x_i x_0 - 2 y_i y_0 + (x_0^2 + y_0^2) + (x_i^2 + y_i^2) = \bar{r}^2 + 2 \bar{r} \tilde{r}_i + \tilde{r}_i^2 \ .$$

Thus,

$$\begin{aligned} & x_i x_0 + y_i y_0 + \tilde{r}_i \bar{r} + (\bar{r}^2 - (x_0^2 + y_0^2))/2 \\ & = ((x_i^2 + y_i^2) - \tilde{r}_i^2)/2 \qquad , \quad i = 1, N \end{aligned} \tag{16}$$

which can be written in matrix form as

$$\begin{bmatrix} x_1 & y_1 & \tilde{r}_1 & 1 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_N & y_N & \tilde{r}_N & 1 \end{bmatrix} V = \begin{bmatrix} (a_1^2 - \tilde{r}_1^2)/2 \\ . \\ . \\ . \\ (a_N^2 - \tilde{r}_N^2)/2 \end{bmatrix} , \tag{17a}$$

where

$$V^T = [\, x_0,\ y_0,\ \bar{r},\ (\bar{r}^2 - a_0^2)/2 \,] \quad , \tag{17b}$$

and ^{T} denotes the transpose of a vector or matrix.

This approach can easily be extended to the location problems in the 3-D Euclidean space and on the sphere (see section IV). Moreover this derivation also provides the interpretation of the parameters b and c:

$$b = -\bar{r} ,$$

$$c = -(\bar{r}^2 - (x_0^2 + y_0^2))/2 .$$

Although $\bar{r}$ is unknown, its expression in terms of the ranges r_i can be fixed if a constraint on the sum of the $\tilde{r}_i$'s is imposed. For example,

$$\sum_{i=1}^{N} \tilde{r}_i = 0 \qquad (18)$$

implies that

$$\bar{r} = \frac{1}{N} \sum_{i=1}^{N} r_i , \qquad (19)$$

i.e. $\bar{r}$ is the average range from the source to the sensors when the set $\{ \tilde{r}_i, i = 1, N \}$ is "normalized" so that (18) holds.

The distinction has been made earlier between two types of techniques for obtaining range difference measurements. In the first case the range estimates $\tilde{r}_i$ are measured and can, therefore, be used in equation (17). In the second case the range differences Δ_{ij} are measured and $\tilde{r}_i$ is not directly available. In the latter case however a generalized TDOA averaging procedure can be used to estimate the $\tilde{r}$'s from the differences Δ_{ji} so that the set $\{\tilde{r}_i = r_i - \bar{r}, i = 1, N\}$ is actually accessible in both cases. This method is described next.

Generalized TDOA Averaging

Given noisy measurements $\Delta_{ji} = [r_i - r_j] + v_{ji}$, many procedures may be devised for estimating the set $\{\tilde{r}_i,\ i = 1,\ N\}$ described above. The method proposed here is basically a least-squares solution, hence some preprocessing has to be done on the Δ's to descard too noisy measurements ("outliers") or separate echoes coming from different sources. This preprocessing may be performed considering the Δ's alone by computing the sums of measured Δ's around several loops defined by the sensors and using those quantities in a hypothesis testing scheme. Alternatively, more sophisticated TDOA estimation techniques can be used which provide an "optimal" set of Δ_{ji}. See, for example, [12] and [13].

Writing the relation (11) for all the pairs of stations yields

$$\underbrace{\begin{bmatrix} \Delta_{21} \\ \Delta_{31} \\ \cdot \\ \cdot \\ \Delta_{N1} \\ \Delta_{32} \\ \cdot \\ \cdot \\ \Delta_{N2} \\ \cdot \\ \cdot \\ \cdot \\ \Delta_{N,N-1} \end{bmatrix}}_{\underline{\Delta}} = \underbrace{\begin{bmatrix} 1 & -1 & 0 & . & . & . & 0 \\ 1 & 0 & -1 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 1 & 0 & 0 & . & . & . & -1 \\ 0 & 1 & -1 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & 1 & 0 & . & . & . & -1 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & . & . & . & 1 & -1 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ r_N \end{bmatrix}}_{r} + \underbrace{\begin{bmatrix} v_{21} \\ v_{31} \\ \cdot \\ \cdot \\ v_{N1} \\ v_{32} \\ \cdot \\ \cdot \\ v_{N2} \\ \cdot \\ \cdot \\ \cdot \\ v_{N,N-1} \end{bmatrix}}_{v}$$

or
$$\underline{\Delta} = P r + v \quad . \qquad (20)$$

Letting

$$1^T = [1, 1, \ldots 1] ,$$

we can write

$$\tilde{r} = r - 1 \bar{r} .$$

Note that if r satisfies (20), then so does $\tilde{r}$. However $\tilde{r}$ will be uniquely determined given the information

$$\underline{\Delta} = P \tilde{r} + v , \qquad (21a)$$

if the constraint (18) is introduced, i.e.

$$1^T \tilde{r} = 0 \quad . \qquad (21b)$$

If the a priori information on the mean of the noise is subtracted from the measured Δ's, v can be considered as zero-mean with covariance matrix R. Making the reasonable assumption that the error v is independent of r, the linear least-squares estimate of $\tilde{r}$ can be found quite easily. The procedure for finding the estimate consists of "whitening" the noise vector v and applying a least-squares algorithm under the constraint represented by equation (18). The details of a sample estimation algorithm are presented in Appendix A.

If we further assume that the errors associated with different TDOA's are independent, in which case the error covariance $R = I$, the least-squares solution of (21) has a particularly simple form. The solution is given by

$$P^T P \tilde{r} = P^T \underline{\Delta} ,$$

that is

$$(NI - 1\,1^T)\,\tilde{r} = \begin{bmatrix} \sum_{j\neq 1} \Delta_{j1} \\ \cdot \\ \cdot \\ \cdot \\ \sum_{j\neq N} \Delta_{jN} \end{bmatrix} . \qquad (22)$$

Using the constraint (18), the linear least-squares estimate of $\tilde{r}$ is, therefore, given by

$$\tilde{r}_i = \frac{1}{N} \sum_{j=1}^{N} \Delta_{ji} \qquad i = 1, N \quad , \qquad (23)$$

which corresponds to simply averaging all of the Δ_{ji} measurements associated with each station S_i. The estimates of the range differences are $\hat{\Delta}_{ji} = \tilde{r}_i - \tilde{r}_j$ with $\tilde{r}_i$ given by (23); their expressions reduce to (12) when $N = 3$. Thus, equation (23) indeed *represents a generalization* of the TDOA averaging method.

Solution Methods

Assuming now that the $\tilde{r}_i$'s have been computed, the set of equations (17) may be used to obtain an estimate of the source location. The geometric configuration of the stations and the source may be such that the accurate determination of the source locations is impossible whatever estimation technique is used. Equation (17) and its extensions to more complicated situations are particularly well suited for displaying such cases (see Appendix B).

When the source is locatable from the TDOA's and the measurement noise is negligible, application of a least-squares algorithm to the equation (17) gives the vector solution V. Furthermore, if a sequence of measurements are available (rather than a single set of $\{\tilde{r}_i\}$), it is possible to use estimation techniques which take into account all measurements to provide the best possible estimate. To see this, we rewrite equation (17) in the form

$$y_t = H_t V_t ,$$

where y is the vector on the right-hand side of (17a) and H is the matrix on the left. The subscript t indicates that these quantities correspond to measurements taken at time t. Assume now that a sequence of measurements of $\{\tilde{r}_i\}$ is available for $t = 0, 1, \ldots$. In estimation terminology we have a sequence of "measurement vectors" y_t and "measurement matrices" H_t, and we want to estimate the unknown "parameter vector" V_t.

Problems of this type have been treated extensively in estimation literature, both for the case where V_t is a constant vector (fixed sources) and for the case where V_t is time varying (moving sources). A number of efficient recursive algorithms for estimating V_t can be found in [14] and [15]. These algorithms can be recursive in time, i.e. when a new data set $\{\tilde{r}_i\}_t$ or equivalently $\{y_t, H_t\}$ becomes available, the estimate of the source location V_t can be updated to give an optimal estimate based on *all past and present* measurements! The algorithms can also be recursive in the number of stations, i.e. given a set of measurements from m stations, they provide an estimate of the source location based on the available data. When a set of measurements from the $m + 1$ station becomes available, the estimate can be updated without recomputing (17). The details of these algorithms will be presented elsewhere.

When the measurements are noisy,

$$\boldsymbol{v}_t = \boldsymbol{H}_t \boldsymbol{V}_t + \epsilon_t \quad ,$$

where ϵ_t represents a discrepancy between the two sides of equation (17). If ϵ_t is assumed to be a sequence of independent random vectors, optimal estimation techniques can still be applied. The situation here is somewhat more complex than the standard linear estimation problem because of the inter-relations between the various components of the vector V. An optimal algorithm for solving equation (17) under these constraints is currently under development and will be presented in a later paper. The algorithm is based on modifications of some nonlinear estimation algorithms of the type derived by Marcus [16] and [17].

IV. Some Extensions

The formulation of the source location problem given in the previous section can be naturally extended in several ways. Here we discuss extensions to three dimensions, to geopositioning on the surface of a sphere and to estimation of source velocity.

3-D Euclidean Space

By writing that the same location is the intersection of the N (hyper-)circles (S_i, r_i) a set of N equations satisfied by the source coordinates and the average range can be obtained for space more complicated than the plan.

A direct extension of the equations obtained in the 2-D case gives,

$$(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2 = (\bar{r} + \tilde{r}_i)^2 , \qquad i = 1, N$$

or

$$-2x_i x_0 - 2y_i y_0 - 2z_i z_0 + (x_0^2 + y_0^2 + z_0^2) + (x_i^2 + y_i^2 + z_i^2) = \bar{r}^2 + 2\bar{r}\tilde{r}_i + \tilde{r}_i^2 .$$

Define a_i by

$$a_i^2 = x_i^2 + y_i^2 + z_i^2 , \qquad i = 0, N .$$

Then the equation above can be written as

$$\begin{bmatrix} x_1 & y_1 & z_1 & \tilde{r}_1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_N & y_N & z_N & \tilde{r}_N & 1 \end{bmatrix} V = \begin{bmatrix} (a_1^2 - \tilde{r}_1^2)/2 \\ \cdot \\ \cdot \\ \cdot \\ (a_N^2 - \tilde{r}_N^2)/2 \end{bmatrix} , \qquad (24)$$

where

$$V^T = [\, x_0,\ y_0,\ z_0,\ \bar{r},\ (\bar{r}^2 - a_0^2)/2 \,] \ .$$

All the comments regarding the solution of (17) are directly applicable here.

Surface of the Sphere

A very similar formulation holds for sources and sensors located on the surface of a sphere. In this case (x_i, y_i, z_i) are direction cosines ($x_i^2 + y_i^2 + z_i^2 = 1$) and r_i measures a distance on the sphere whose radius is taken as the unit of length, hence r_i is an angle. In this case it is easy to see that the following equations hold,

$$x_0\, x_i + y_0\, y_i + z_0\, z_i = \cos r_i = \cos \bar{r} \cos \tilde{r}_i - \sin \bar{r} \sin \tilde{r}_i \ , \qquad i = 1, N \quad (25a)$$

which can be rewritten in matrix form as,

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_N & y_N & z_N \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} \cos \tilde{r}_1 & -\sin \tilde{r}_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cos \tilde{r}_N & -\sin \tilde{r}_N \end{bmatrix} \begin{bmatrix} \cos \bar{r} \\ \sin \bar{r} \end{bmatrix} \ . \qquad (25b)$$

Velocity Determination

The above approach may be used to determine the components of the velocity vector of a moving source when, in addition to the TDOA's, derivatives of the range differences are available (e.g. from FDOA's, frequency differences of arrival). Then, applying to the $\dot{\Delta}_{ji}$'s the same generalized averaging procedure as for the range differences, the set of derivatives $\{\dot{\tilde{r}}_i = \dot{r}_i - \dot{\bar{r}},\ i = 1,\ N\,\}$ may be estimated. It may also happen that the $\dot{\tilde{r}}_i$ are directly available.

Differentiation of the equations of the circles $(S_i,\ r_i)$ yields, in the 2-D Euclidean case, assuming the stations are fixed:

$$2\,\dot{x}_0\,(x_i - x_0) + 2\,\dot{y}_0\,(y_i - y_0) = 2\,(\dot{\bar{r}} + \dot{\tilde{r}}_i)\,(\bar{r} + \tilde{r}_i)\ , \qquad i = 1, N$$

which can be written as

$$\begin{bmatrix} x_1 - x_0 & y_1 - y_0 & -(\bar{r} + \tilde{r}_1) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_N - x_0 & y_N - y_0 & -(\bar{r} + \tilde{r}_N) \end{bmatrix} \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{\bar{r}} \end{bmatrix} = \begin{bmatrix} (\bar{r} + \tilde{r}_1)\,\dot{\tilde{r}}_1 \\ \\ \cdot \\ \cdot \\ (\bar{r} + \tilde{r}_N)\,\dot{\tilde{r}}_N \end{bmatrix}. \qquad (26)$$

Hence, having solved (17) for x_0, y_0, $\bar{r}$, (26) can be solved for $\dot{x}_0$, $\dot{y}_0$, $\dot{\bar{r}}$. Moreover, if an initial estimate of the source location is available, it is theoretically possible, by integration of x_0 and y_0, to obtain the location and velocity components of the source from rate of range difference measurements only. We are currently investigating the proper problem formulation for determining source location from FDOA measurements only.

V. Conclusions

The source location problem has many aspects, only a few of which have been discussed in the previous sections. Here we shall briefly describe two other facets of this problem which we are investigating:

- Distributed algorithms for source location. In some applications (e.g. [2]) it is important to perform the source location calculations in a distributed fashion. Each sensor has An associated processor which can perform some of the computations, which are then transmitted to a central processor where the partial results are combined to provide the global optimal estimate. A distributed implementation of the computations minimizes the communication requirements between sensors and leads to flexible and reliable system designs. The linearity of equation (17) makes such distributed implementation much easier than in the case of having a set of nonlinear equations. A more detailed explanation of the distributed processing aspects of our approach can be found in [19].

- Optimal location estimation. The process of finding the source location consists basically of two steps: first, estimating a set of TDOA's (Δ_{ij}) or travel times ($\tilde{r}_i$) from sensor measurements, using the TDOA averaging method or a more sophisticated estimation scheme. Second, use this set of estimates to compute the source location via equation (17) or some other approach. A natural question is whether this two-step approach is optimal. In general we cannot expect the optimal estimator to have this structure. The complexity of the nonlinear equations found in the hyperbolic LOP approach prevented an adequate treatment of this important issue. The much simpler structure of our approach makes the problem of finding an optimal estimator for the source coordinator more tractable. This problem is currently under investigation, using a combination of the framework described in this paper and a new approach to the TDOA estimation problem described in [13].

We are still in the process of evaluating the performance of the proposed approach and comparing it to alternative techniques. However, we believe that the framework presented here gives a particularly convenient way of analyzing and solving the source location problem, and all the indications are that its performance will be at least as good as of currently used techniques in the low noise case. In the case of very noisy measurements (low signal-to-noise ratios) significant improvements are expected since optimal estimation algorithms can be applied, whereas only ad-hoc and sometimes inconsistent approaches have been used in the past.

References

[1] R.O. Schmidt, "A New Approach to Geometry of Range Difference Location," *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-8, No. 6, Nov. 1972, pp. 821-835.

[2] M. Morf, B. Friedlander and J. Newkirk, "Tutorial Survey of Algorithms for Locating and Identifying Spatially Distributed Sources and Receivers," *Proc. Distributed Sensor Nets Workshop*, Carnegie-Mellon Univ., Pittsburgh, Penn., Dec. 1978, pp. 94-104.

[3] 1971 Pioneer Award for Robert L. Frank and Winslow Palmer, Part III, Cytac, Loran C and Loran D, *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-7, no. 5, Sept. 1971, pp. 1018-1021.

[4] J.A. Pierce, "Omega," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-1, no. 3, Dec. 1965, pp. 206-215.

[5] H. Shoemaker, "The NAVSTAR/Global Positioning System," *Proc. NAECON '75 Conf.*, Dayton, Ohio, June 1975.

[6] H.L. Groginsky, "Position Estimation Using Only Multiple Simultaneous Range Measurements," *IRE Trans. Aeronautical and Navigational Electronics*, Sept. 1959, pp. 178-187.

[7] J.L. Poirot and G.V. McWilliams, "Application of Linear Statistical Models to Radar Location Techniques," *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-10, No. 6, Nov. 1974, pp. 830-834.

[8] W.M. Foy, "Position-Location Solutions by Taylor-Series Estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-12, no. 2, March 1976, pp. 187-194.

[9] J.P. Van Etten, "Navigation Systems: Fundamentals of Low and Very-low Frequency Hyperbolic Techniques," *Electrical Commun.*, vol. 45, no. 3, 1970, pp. 192-212.

[10] H.B. Lee, "A Novel Procedure for Assessing the Accuracy of Hyperbolic Multilateration Systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-11, no. 1, Jan.1975, pp. 2-15.

[11] N. Marchand, "Error Distributions of Best Estimate of Position from Multiple Time Difference Hyperbolic Networks," *IEEE Trans. Aerosp. Navig. Electron.*, vol. ANE-11, no. 2, June 1964, pp. 96-100.

[12] C.H. Knapp and G.C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-26, no. 6, Aug. 1976, pp. 320-327.

[13] B. Friedlander and M. Morf, "A System Identification Approach to Locating Spatially Distributed Sources," submitted to the 18th IEEE Conference on Decision and Control.

[14] A. Gelb, *Applied Optimal Estimation*, MIT Press, Cambridge, Mass., 1974.

[15] C.L. Lawson and R.J. Hanson, *Solving Least Squares Problems*, Prentice Hall, Inc., New York, 1974.

[16] S.I. Marcus, "Optimal Finite Dimensional Recursive Estimators for Discrete-Time Stochastic Nonlinear Systems," *Proc. 15th Allerton Conf. Co[illegible]st. Computation*, Monticello, Illinois, Sept. 1977, pp. 52-61.

[17] S.I. Marcus, S.K. Mitter, D. Ocone, "Finite Dimensional Nonlinear Estimation in Continuous and Discrete Time," *Int. Conf. on Analysis and Optimization of Stochastic Systems*, Oxford, Sept. 1978.

[18] J.R. Harris, Tracor Inc., personal communication.

[19] E. Verriest, B. Friedlander and M. Morf, "Distributed Processing

in Estimation and Detection," submitted to the 18th IEEE Conference on Decision and Control.

Appendix A: Computation of the Averaged Range Differences

Equations

Given a set of range differences

$$\Delta_{ji} = r_i - r_j + v_{ji} \qquad i = 1, N \quad , \quad j = i+1, N$$

or, using matrix notation,

$$\underline{\Delta} = \boldsymbol{P}\boldsymbol{r} + \boldsymbol{v} \quad ,$$

where $\boldsymbol{v}$ is zero-mean with covariance R. The averaged range differences are obtained as $\hat{\Delta}_{ji} = \tilde{r}_i - \tilde{r}_j$, where $\tilde{\boldsymbol{r}}$ is the least-squares solution of $\boldsymbol{P}\tilde{\boldsymbol{r}} = \Delta$ which satisfies the constraint $\boldsymbol{1}^T \tilde{\boldsymbol{r}} = 0$. The description of an algorithm for finding $\tilde{\boldsymbol{r}}$ is the object of this Appendix. For a more detailed explanation of least-squares techniques for solving linear equations see [15].

Algorithm

First the noise $\boldsymbol{v}$ is whitened:

$$\boldsymbol{R}^{-1/2}\,\underline{\Delta} = \boldsymbol{R}^{-1/2}\,\boldsymbol{P}\,\tilde{\boldsymbol{r}} + \boldsymbol{w} \quad , \qquad \text{(A1)}$$

where $\boldsymbol{w} = \boldsymbol{R}^{-1/2}\,\boldsymbol{v}$ is white. Then introduction of the constraint on $\tilde{\boldsymbol{r}}$ yields the system

$$\begin{bmatrix} \boldsymbol{1}^T \\ \boldsymbol{R}^{-1/2}\,\boldsymbol{P} \end{bmatrix} \tilde{\boldsymbol{r}} = \begin{bmatrix} 0 \\ \boldsymbol{R}^{-1/2}\,\underline{\Delta} \end{bmatrix} , \qquad \text{(A2)}$$

where the first equation must be strictly satisfied and the other ones only in the least-squares sense.

This system is equivalent to

$$\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{b} \quad , \qquad \text{(A3)}$$

where

$$A = \begin{bmatrix} -N^{1/2} & 0 \; . \; . \; . \; 0 \\ \multicolumn{2}{c}{R^{-1/2}\, P\, H} \end{bmatrix} \qquad (1 + (N(N-1))/2) \;\; \mathrm{x} \;\; N$$

$$s = \quad H\,\tilde{r} \qquad\qquad N \;\mathrm{x}\; 1$$

$$b = \begin{bmatrix} 0 \\ \\ R^{-1/2}\,\underline{\Delta} \end{bmatrix} \qquad (1 + (N(N-1))/2) \;\mathrm{x}\; 1 \quad ,$$

and H is the Householder reflection defined by the vector $\mathbf{1}$:

$$H = I_N - \frac{1}{u_1}\, \mathbf{u}\,\mathbf{u}^{\mathrm{T}},$$

where

$$\mathbf{u}^{\mathrm{T}} = [\, 1 + N^{-1/2},\; N^{-1/2},\; .\; .\; .\; .,\; N^{-1/2} \,] \qquad 1 \;\mathrm{x}\; N \;.$$

The N - 1 last columns of A define a Householder transformation Q , a product of several Householder *reflections, such that*

$$Q\,A = \left[\begin{array}{c:c} -N^{1/2} & 0 \;\; . \;\; . \;\; . \;\; 0 \\ \hdashline & \\ & U \\ q & \\ \cdashline{2-2} & O \end{array}\right] \quad ,$$

where U is $(N-1)$ x $(N-1)$ and upper triangular, and

$$Q\,b = [\; O,\;\; p_1{}^{\mathrm{T}},\;\; p_2{}^{\mathrm{T}} \;]^{\mathrm{T}} \quad ,$$

with p_1 : $(N-1)$ x 1 and p_2 : $((N-1)(N-2))/2$ x 1 . The least-squares solution of (A3) is then readily obtained as:

$$s_1 = 0 \qquad\qquad \text{(a constraint)},$$

and the other components of s are found by back-substitution from

$$U \bar{s} = p_1 \quad ,$$

where,

$$\bar{s}^T \hat{=} [s_2, \ldots , s_N] \quad . \qquad \text{(A4)}$$

The components of the least-squares solution $\tilde{r} = H s$ are:

$$\tilde{r}_1 = -\sigma$$

and

$$\tilde{r}_i = s_i - \sigma (1 + N^{1/2})^{-1} \quad , \quad i = 2, N \qquad \text{(A5)}$$

with

$$\sigma = N^{-1/2} \sum_{i=2}^{N} s_i \quad .$$

The corresponding minimum-squared error is given by:

$$m.s.e. = p_2^T p_2 \qquad \text{(A6)}$$

Computational Complexity

The quantities Q, $R^{-1/2}$ and U may be precomputed so that, when no measurement is rejected, the computations reduce to:

i) multiply $Q R^{1/2}$ and $\underline{\Delta}$ to obtain p_1 ($(N(N-1)^2)/2$ multiplications)

ii) solve (A4) ($(N(N-1))/2$ multiplications and $(N-1)$ divisions)

iii) get the $\tilde{r}_i$'s from (A5)

iv) optional:

compute P_2 ($(N(N-1)^2(N-2))/4$ multiplications)

apply (A6) to evaluate the m.s.e. ($((N-1)(N-1))/2$ multiplications).

In summary, $(N^2(N-1))/2$ multiplications and $(N-1)$ divisions are needed for $\tilde{r}$, plus approximately $N^4/4$ multiplications for the m.s.e. (If $\boldsymbol{R} = \boldsymbol{I}$, then $\boldsymbol{p}_2^T \boldsymbol{p}_2 = \underline{\Delta}^T \underline{\Delta} - \boldsymbol{p}_1^T \boldsymbol{p}_1$ and the m.s.e. is evaluated in $((N+2)(N-1))/2$ multiplications)

When some measurements are rejected, more computations are involved because the precomputed matrices cannot be used with only simple transformations. For example, the inverse square-root of the reduced covariance matrix cannot, in general, be directly obtained from the precomputed $\boldsymbol{R}^{-1/2}$ by simply crossing out some rows and columns.

Appendix B: Conditioning of the Source Location Problem (Locatability)

According to the type of measurement used to estimate position (e.g. range measurement or bearing angle measurement), there exist some particular configurations of the source and the stations for which the estimated source location is very sensitive to measurement errors. This phenomenon, known as "geometric dilution of precision" [8] - [10], also appears when range differences are measured; the equations derived in sections III and IV prove particularly helpful to determine the situations for which it occurs for source location in the 2-D and 3-D Euclidian spaces and on the surface of the sphere.

The set of equations (17) will have an infinite number of solutions if the rank of the matrix

$$\begin{bmatrix} x_1 & y_1 & \tilde{r}_1 & 1 \\ : & : & : & : \\ : & : & : & : \\ x_N & y_N & \tilde{r}_N & 1 \end{bmatrix}$$

is less than 4. This is equivalent to saying that the rank of the matrix

$$\begin{bmatrix} x_1 & y_1 & r_1 & 1 \\ : & : & : & : \\ : & : & : & : \\ x_N & y_N & r_N & 1 \end{bmatrix}$$

is less than 4, since $\tilde{r}_i = r_i + \bar{r}$. This means that there exists $(a, b, c, d) \neq 0$ such that for every receiver the triple (x, y, r) satisfies

$$a x + b y + c r + d = 0 . \qquad (B1)$$

Consider now several special cases:

i) If $c = 0$, all the receivers are on the same line $a\,x + b\,y = -d$.

Assume now that $c \neq 0$. Letting

$$D = d + a\,x_0 + b\,y_0 \quad ,$$

(B1) becomes

$$a\,(x - x_0) + b\,(y - y_0) + c\,r + D = 0$$

and, dividing by r, we get

$$a\cos\theta + b\sin\theta + c + \frac{D}{r} = 0 \ .$$

ii) If $D \neq 0$ we obtain

$$\frac{r}{D} = -\frac{1}{c + a\cos\theta + b\sin\theta}$$

i.e.

$$r = \frac{p}{1 - e\cos(\theta - \phi)} \quad , \qquad \text{(B2)}$$

where

$$p = -\frac{D}{c}$$

$$e = \left(\frac{a^2 + b^2}{c^2}\right)^{1/2}$$

$$\cos\phi = -\frac{a}{(a^2 + b^2)^{1/2}}$$

$$\sin\phi = -\frac{b}{(a^2 + b^2)^{1/2}} \ .$$

Thus, all stations are on a general conic with the source as one focus, as stated in [1].

iii) If $D = 0$ all the stations are on the part defined by the set of coordinates

$$\{\,(x,y){:}\ sign\,[\,a\,(x - x_0) + b\,(y - y_0)\,] = -\,sign\ c\,\}$$

of a general conic whose center is the source. (The conic may degenerate into two lines intersecting at the source location).

3-D Euclidean Space

Clearly if (24) is considered, there is degeneracy in the following situations

- all the stations are in the same plane (a 2-D subspace)
- all the stations are on a general quadric surface with the source as one focus
- all the stations are on a well defined part of a general quadratic (which may degenerate into two planes) whose center is the source.

Surface of the Sphere

Letting

$$M = \begin{bmatrix} x_1 & y_1 & z_1 & -\cos\tilde{r}_1 & -\sin\tilde{r}_1 \\ : & : & : & : & : \\ : & : & : & : & : \\ x_N & y_N & z_N & -\cos\tilde{r}_N & -\sin\tilde{r}_N \end{bmatrix}$$

equation (25) may be rewritten as

$$M \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \cos\bar{r} \\ \sin\bar{r} \end{bmatrix} = 0 \quad ,$$

with

$$x_0^2 + y_0^2 + z_0^2 = 1$$
$$\cos^2\bar{r} + \sin^2\bar{r} = 1 \quad . \qquad \text{(B3)}$$

Obviously if $\rho(M) = 5$ the system does not have any solution (ρ denotes the matrix rank).

If $\rho(M) = 4$ there exists a vector

$$t = [x, y, z, u, v]^T \quad ,$$

such that

$$M\, t = 0 \; .$$

However, if

$$x^2 + y^2 + z^2 \neq u^2 + v^2 \; ,$$

(B3) will not have a solution. Now if M arises from a set of perfect data for N stations then $\rho(M) < 5$; if $\rho(M) = 4$ then (B3) has two solutions and only two, one being the opposite of the other.

Degeneracy arises only if $\rho(M) < 4$. Thus conditions for degeneracy are obtained by determining the configurations for which the source and $N \geq 4$ stations give an M such that $\rho(M) = 3$. Let

$$u = \cos r$$

and

$$v = \sin r \quad ,$$

then $\rho(M) = 3$ is equivalent to the existence of two 5-tuples (a, b, c, d, e), (a', b', c', d', e') such that:

$$\begin{aligned} a\,x + b\,y + c\,z + d\,u + e\,v &= 0 \\ a'x + b'y + c'z + d'u + e'v &= 0 \; . \end{aligned} \qquad \text{(B4)}$$

If

$$\det \begin{bmatrix} d & e \\ d' & e' \end{bmatrix} \neq 0 \qquad ,$$

then

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} d & e \\ d' & e' \end{bmatrix}^{-1} \begin{bmatrix} a & b & c \\ a' & b' & c' \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\hat{=} \begin{bmatrix} A & B & C \\ A' & B' & C' \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad . \qquad \text{(B5)}$$

Hence u can be interpreted as the cosine of the distance to the point (A, B, C) and v as the cosine of the distance to the point (A', B', C'). Thus $(1 - v^2)^{1/2}$ is also the *cosine* of the distance to the great circle whose polar point is (A', B', C'). But $(1 - v^2)^{1/2}$ is u, therefore the two conditions (B5) are equivalent to saying that the distance to the point (A, B, C) is equal to the distance to the great circle with polar point (A', B', C') and that is the definition of some type of spherical conic . Moreover, since $u = \cos r$, the source location is at the point (A, B, C), one focus of the conic.

There are still some other cases of degeneracy, not given here.

Domain of Unlocatability

When the configuration of the stations and source is close to degeneracy the solution is very sensitive to the noise in the data whatever the procedure used to find this solution. The determination of the domain of unlocatability of the source given a configuration of the stations is therefore of basic interest.

Considering the problem of location in the plane, if the N stations are on a straight line the domain of unlocatability is the whole plane. If $N = 4$ and the stations are not aligned then there is a pencil of conics going through the stations and part of the unlocatability domain is the locus of the two foci of these conics. If

$N = 5$ there will exist in general a conic going through the stations locations so that the unlocatability domain will consist of the two foci of this conic.

If $N > 5$ one may try to define which configuration(s) of the stations optimizes the conditioning of the problem; however if N is very large a deterministic optimization is not necessary. A random scattering of the stations is sufficient since it can be shown (using the theory of random matrices) that for sensors distributed at random and independently in the space, the probability of poor conditioning of the equations obtained from N sensors (defined as the sum of the distances of the sensors to the closest configuration of degeneracy being less than some threshold) follows asymptotically a Poisson law whose mean is an increasing function of the density of the sensors and of the threshold. Hence for large N the probability of near degeneracy when the stations are randomly distributed is extremely small.

DISTRIBUTED PROCESSING IN ESTIMATION AND DETECTION[‡]

E. Verriest,[*] B. Friedlander[**] and M. Morf[*]

Abstract

The need for decentralized computation arises in many problems and applications. One of the prime motivations is to try to keep the communication load between different sites or even between different local processors down. In the case of a distributed sensor network (DSN) an enormous amount of raw data is generated by the sensors, generally distributed over some area. In this paper we shall have a closer look at some distributed forms for optimal estimation and detection problems that arise in such a network.

A proposed batch algorithm is believed to be new, as well as a step-by-step algorithm when sensor parameters are a priori unknown. It has the added feature that it remains suboptimal under sensor failure. The basic questions are: What information should be exchanged? How should it be combined?

[*] Information Systems Laboratory, Stanford University, Stanford, CA 94305.

[**] Systems Control, Inc., Palo Alto, CA 94304.

[‡] This work was supported by the Defense Advanced Research Projects Agency through contract MDA 903-78-C-0179, and the facilities at the Stanford Artificial Intelligence Laboratory.

1. INTRODUCTION

The need for decentralized computation arises in many problems and applications, such as power networks [1], distributed sensor networks [2] and others, see, e.g. [3]. Briefly we can mention that one of the prime motivations is to try to keep the communication-load between different sites, or even between different local processors down. We assume that there is a limited communications bandwidth, or that the communication costs are relatively high compared to computation costs. In the case of a distributed sensor network (DSN) an enormous amount of raw data is generated by the sensors, generally distributed over some area. The different sensor sites are envisioned to be connected via a computer network. Not all of the data may be relevant; furthermore, many of the computations with the data involve only parameters and data related to one 'station'. Therefore such operations can be performed by a processor near or at the sensor site, and only some preprocessed results need to be communicated to the central station, or a user on the network.

The detection and tracking of low flying aircraft and cruise missiles is one example where distributed sensor networks are of interest. The sensors can be active or passive, but the generated amount of data is large compared to the number of parameters or states to be estimated.

When we have a completely centralized scheme, i.e., all the data is communicated noise free from the sensors to one location, then it is well known that the minimum error-variance state estimator is given by the Kalman-filter for linear models.

Other applications - as in power systems - deal with the state estimation of a large number of interconnected subsystems. Many attempts have been made in the past to decentralize the Kalman-filter, and both optimum and sub-optimum algorithms resulted [3].

In this paper we shall have a closer look at the possible distributed forms for optimal estimation and detection problems. The basic questions are: What information should be exchanged? How should it be combined.

Section 2 handles the simplest case of parameter estimation, or systems with no dynamics, introducing the basic tools in more detail and providing the most insights in the distributed aspects.

Section 3 extends these results to systems with dynamics, e.g., tracking problems, for which an optimal step-by-step update of the estimates is demanded (i.e., getting real time estimates).

This requires substantial communication, and the problem was treated by Speyer [4] and others [5,6]. However, the algorithm proposed by Speyer implies the knowledge of the measurement parameters $(H^{(i)},R^{(i)})$ of each substation beforehand. This is an unlikely situation in many applications, for instance when the sensors are mobile and the measurement parameters dependent on the coordinates of the node. We present here an alternative algorithm that does not require this a priori knowledge but by necessity then results in a higher number of data to be transmitted. In Section 4, we shall look at a distributed batch-processing algorithm. This is appropriate if one does not need estimates at every time-point. We give a development of such optimal "block" processing based on our scattering approach [7] in order to gain more insights in the various possible forms of distributing computations and exchanging information, i.e., measured data or partial estimates.

A brief discussion of the static and dynamic detection problems is presented in Section 5. Features of the algorithms are reconfigurability, reduced communication requirements and survivability (i.e., robust, suboptimal by partial breakdown). Flexibility from the user's standpoint can be incorporated in a hierarchical system design (local regions, area coordinates, etc.).

2. PARAMETER ESTIMATION

This discussion is focused on algorithms that lend themselves naturally to distributed processing. We shall first consider the static estimation problem, since it leads to the simplest example of a distributed estimation algorithm, and will therefore give the most insight into the problem of distributed processing. These results will be used in Section 3.

In this problem, x is an n-dimensional vector of unknown parameters, which we would like to estimate. We will assume that all the information that we have about x are some (noise-corrupted) measurements, and some a priori information. Suppose that the uncertainty in this a priori is specified in terms of the second moment of x. The prior information about x is then given by

$$E[x] = x_o \tag{2.1}$$

$$E[\tilde{x}_o \tilde{x}_o'] = \Pi \tag{2.2}$$

where E denotes expected value. In the rest of this paper, we assume for simplicity that $x_o = 0$. The measurements are situated at disjoint locations N_i, $i = 1,\ldots,r$. This may be disjoint in space (disjoint sensors) as well as in time (as is the case in batch processing of repeated measurements), or a combination of the two. We do not have to discriminate between space and time here, because in this problem, no dynamics are involved, and therefore, there is no time evolution. We shall refer to N_i as 'station i'.

Let N_i have p_i measurements, modeled as:

$$y^{(i)} = H^{(i)}x + v^{(i)} \quad ; \quad i = 1,\ldots,r \tag{2.3}$$

where H_i is a $p_i \times n$ matrix, denoting an 'observation matrix' that depends on the node (or station) N_i. The term v_i represents the measurement error, and is modeled as an 'observation noise'. We shall assume that the p_i measurements at N_i are unbiased, i.e., $E\, v^{(i)} = 0$, and that further the observations are uncorrelated, and 'reduced' (i.e., have unit variance) $E\, v^{(i)} v^{(i)'} = I$. These assumptions entail no loss of generalization, since we can always preprocess the data for an arbitrary measurement error variance, provided that this variance is nonsingular. Indeed, let in general $E\, v^{(i)} v^{(i)'} = R_v$, then $z_i = R_v^{-1/2}\, y_i$ are the 'reduced' observations, because then:

$$z_i = \left(R_v^{-1/2}\, H^{(i)}\right) x + R_v^{-1/2}\, v^{(i)} \overset{\Delta}{=} \bar{H}^{(i)}\, x + \bar{v}^{(i)} \quad (2.4)$$

where

$$E\, \bar{v}^{(i)} = R_v^{-1/2}\, E\, v^{(i)} = 0 \quad (2.5)$$

$$E\, \bar{v}^{(i)} \bar{v}^{(i)'} = R_v^{-1/2}\, E\, v^{(i)} v^{(i)}\, R_v^{-T/2} = I\ . \quad (2.6)$$

It is clear that this preprocessing involves only one station, and therefore can be done locally, without any communication with other locations. Also $v^{(i)}$ is assumed to be uncorrelated with the parameter x, and all measurements performed at stations N_j, $j \neq i$.

With respect to a spatial distribution of the N_i, it means that the measurement errors at one station are uncorrelated with the errors at another station, in most cases, a quite reasonable assumption. If we consider temporal distribution (batch processing) it means that the measurement errors in one batch are uncorrelated with those in previous batches. For the moment, we will leave the idea of distributed organization, and present the optimal solution without decentralized considerations.

Define:

$$y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(r)} \end{bmatrix} , \quad H = \begin{bmatrix} H^{(1)} \\ \vdots \\ H^{(r)} \end{bmatrix} , \quad v = \begin{bmatrix} v^{(1)} \\ \vdots \\ v^{(r)} \end{bmatrix} \qquad (2.7)$$

so that the 'global observation' equation becomes:

$$y = Hx + v \qquad (2.8)$$

It is now well known (e.g., see [8,9]) that the linear-least-mean-square estimate of x given y is given by:

$$\hat{x} = (E\ xy')(E\ yy')^{-1}\ y \qquad (2.9)$$

By direct verification, it is clear that:

$$E\ yy' = H\ \Pi\ H' + I$$

$$E\ xy' = \Pi\ H'$$

So:

$$\hat{x} = \Pi\ H'(H\ \Pi\ H' + I)^{-1}\ y \qquad (2.10)$$

After combining the a priori with the observations, the remaining uncertainty in x is given by:

$$P = E(x-\hat{x})(x-\hat{x})'$$

which after some algebra yields:

$$P = \Pi - \Pi\ H'(H\ \Pi\ H' + I)^{-1}\ H\Pi \qquad (2.11)$$

Using the matrix identity

$$(A+BCD)^{-1} = A^{-1}-A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}$$

we obtain (2.10) and (2.11)

$$\hat{x} = \left(\Pi^{-1} + \sum_{i=1}^{r} H^{(i)\prime}H^{(i)}\right)^{-1} \sum_{i=1}^{r} H^{(i)\prime}y^{(i)} \qquad (2.12)$$

and

$$P = \left(\Pi^{-1} + \sum_{i=1}^{r} H^{(i)\prime}H^{(i)}\right)^{-1} . \qquad (2.13)$$

We can now compare the use of formulas (2.10) and (2.11) with those of (2.12) and (2.13).

In case the observation matrices are known a priori by the central processor, then each station only needs to send the observed data, or its equivalent. Under the assumed knowledge of H, the posterior error covariance P can be precomputed by either (2.11) or (2.12). The story is different for the estimates. Either each node transmits its p_i data to the central station, which combines these according to the formula (2.10) or each station precomputes a "partial estimate" $z^{(i)} \overset{\Delta}{=} H^{(i)\prime} y^{(i)}$ (or $PH^{(i)\prime} y^{(i)}$ in case all nodes have a "copy" of the observation model (2.8)).

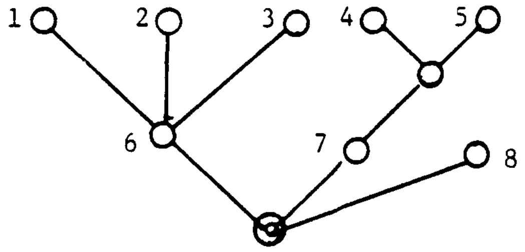
The first method results in $\sum_{i=1}^{r} p_i \overset{\Delta}{=} p \cdot r$ data transmissions (p is the average number of observations per station) while the decentralized computation scheme requires only nr data transmissions. Savings in communications are thus substantial in a many-observation situation (i.e., $p > n$). This is even more so if the observation parameters from each node have to be communicated. As indicated by the summation over i, the formulas (2.12) and (2.13) lead naturally to a distributed scheme.

Now, next to computing the $z^{(i)}$, each node N_i also forms the matrix $L^{(i)} = H^{(i)\prime} H^{(i)}$ of dimension n x n. Then $z^{(i)}$ and $L^{(i)}$ are transmitted to the central processing station. Note that because of the symmetry of $L^{(i)}$, there are only $\frac{n(n+1)}{2}$ independent quantities. Thus node N_i transmits $n + \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{3}{2} n$ numbers to the central processor instead of p_i (from the data) and np_i from $H^{(i)}$. The central processor then adds the $L^{(i)}$ to Π^{-1}, computes this inverse, and post-multiplies by the sum of the $z^{(i)}$.

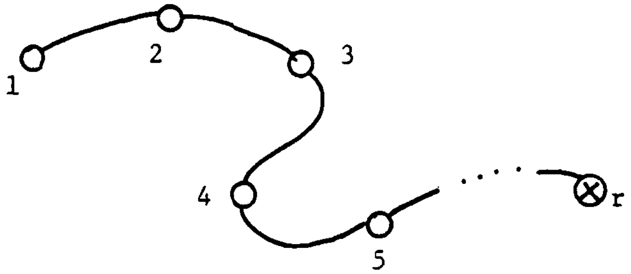
In the case of a spatially distributed system, we have savings in data transmission with distributed processing (versus centralized) if $p_i > \frac{n}{2} + \frac{n}{n+1}$ which can be simplified to $p_i \geq \frac{n}{2} + 1$ (since n and p_i are integers).

The total over all r stations involves: $r\frac{n^2+3n}{2}$ transmissions for the distributed scheme, versus $\sum_{i=1}^{r} (n+1)(p_i = (n+1)p$ for the transmission of the data to a central processing unit. Again the trade-off is: $r\frac{n^2+3n}{2} \leq (n+1)p$ or roughly: $\frac{p}{r} > \frac{n}{2} + 1$. If we consider the batch processing, the above holds if we substitute 'number of transmissions' by 'number of memory locations.' Of course, computational costs and transmission costs may be weighted differently, so that the trade-off between communication load and the computational burden at a station may be expected at another value of p/r.

There are very many details that it will be necessary to deal with in any actual implementation of an algorithm such as the one described. To mention a few of the possible refinements, we can, instead of having all r stations transmitting to the central processor, have a tree-type communication network, e.g.,



or, even simpler, a 'linear'structure



In the latter, all stations compute their $z^{(i)}$ and $L^{(i)}$, but now, station N_2 waits for transmission until it receives $z^{(1)}$ and $L^{(1)}$ and computes

$$\bar{z}^{(2)} = z^{(1)} + z^{(2)}$$
$$\bar{L}^{(2)} = L^{(1)} + L^{(2)}$$

Now it sends $\bar{z}^{(2)}$ and $\bar{L}^{(2)}$ across to (3) instead of $z^{(1)}, z^{(2)}$, $L^{(1)}, L^{(2)}$, thus 'compressing' the data. Clearly, there are again $\frac{r\ n(n+3)}{2}$ transmissions needed, but depending on the sequencing, the distances involved might be shorter, and also more processing goes on at the nodes 2 to r, and less at the central coordinator. In case of a temporal distribution, this is even clearer. From the first batch $z^{(1)}$ and $L^{(1)}$ are computed and put into memory (the line in the figure denotes now temporal evolution rather than geographical routing). When the second batch is processed, the $z^{(2)}$ and $L^{(2)}$ are added to the contents of the memory, and therefore can be stored in the same memory unit. Appendix A describes a further refinement for the computation and possible reduction in the transmission of $L^{(i)}$ using square-root methods.

3. THE DYNAMIC ESTIMATION PROBLEM: REAL TIME UPDATES

3.1 Problem Formulation

We consider here the state estimation of a linear dynamic system. We shall only be concerned with the discrete-time formulation of the problem, undoubtedly the most practical for digital computer applications. We assume that the knowledge about the initial state of the system can be modelled with a Gaussian distribution, as we did in the previous section (the a priori knowledge of x). The inputs to the system are supposed to be purely random, uncorrelated with the initial state (x_0), and are modelled as "white noise."

Let the state model be:

$$x_{t+1} = F_t x_t + u_t \tag{3.1}$$

with x_t the unknown n-dimensional state at time t, under the assumptions:

$$x_0 \sim N[\bar{x}_0, \Pi_0] \tag{3.2}$$

$$u_t \sim N[0, Q_t] \tag{3.3}$$

and

$$E[u_t\ u_s'] = 0 \quad \text{if} \quad t \neq s \tag{3.4}$$

$$E[u_t\ x_0'] = 0 \quad \text{for all} \quad t \quad . \tag{3.5}$$

The nodal observations form a time-sequence, where at each time instant t we have:

$$y_t^{(i)} = H_t^{(i)} x_t + v_t^{(i)} \quad ; \quad i = 1,\dots,r \quad . \qquad (3.6)$$

For generality we consider time variable observation noise characteristics

$$E\, v_t^{(i)} = 0$$

$$E\, v_t^{(i)} v_s^{(i)} = R_t^{(i)} \delta_{ts} \quad . \qquad (3.7)$$

A special feature of the decentralization of the sensors is the fact that the measurements are completely independent and therefore that the noise at different nodes are independent:

$$E\, v_t^{(i)} v_s^{(j)} = 0 \quad \forall \; t,s \text{ if } i \neq j \quad . \qquad (3.8)$$

The last assumption that we shall make is the uncorrelatedness of plant- and observation noise:

$$E\, v_t^{(i)} u_s' = 0 \quad \forall \; t,s; \; \forall i \quad . \qquad (3.9)$$

Defining the quantities y_t, H_t, and v_t for each time instant as in (2.7), then we get the time-indexed equation (2.8):

$$y_t = H_t x_t + v_t \quad . \qquad (3.10)$$

By virtue of (3.7) we have the combined noise characteristics:

$$E\, v_t = 0$$

$$E\, v_t v_s' = \text{Blockdiagonal } [R_t^{(1)} \dots R_t^{(r)}] \delta_{t,s} . (3.11)$$

The optimal solution of this estimation problem is given by the well-studied Kalman filter (e.g., see [8]). This gives a step-by-step up-dating method which incorporates a "time-update" and a "measurement up-date."

3.2 Decentralized Estimation with Known Observation Parameters $H_t^{(i)}, R_t^{(i)}$

In the context of stochastic control, Speyer [4] recently gave a method for a distributed optimal step-by-step solution. Since there the goal is to make a robust optimal decentralized controller, a "best" estimate is necessary at each node N_i. In case a node is isolated from the network, a "local" Kalman filter gives a local (sub)optimal estimate, and this is used for the state feedback controller. With the network available, additional outside information is used to upgrade the estimate. In order to accomplish this, Speyer's algorithm requires each node to compute another n-dimensional vector based on its local data. The local estimate and the additional vector are released to the network and all nodes or a central node that have access to these $2n \cdot r$ parameters can combine these to find the optimal estimate (thus coinciding with the estimate, computed from a "global" Kalman filter).

For completeness we give here the algorithm in a slightly modified fashion, but equivalent to the original. The details of the derivation can be found in [4].

First, each node computes the local estimate, based on $Z_t^{(i)}=\{y_0^{(i)},\ldots,y_t^{(i)}\}$ by a local Kalman filter:

$$\hat{x}^{(i)}_{t+1|t} = F_t \hat{x}^{(i)}_{t|t-1} + K_t^{(i)}(y_t^{(i)} - H_t^{(i)} \hat{x}^{(i)}_{t|t-1}) \qquad (3.12)$$

$$\hat{x}^{(i)}_{0|-1} = \bar{x}_0$$

$$K^{(i)}_t = F_t P^{(i)}_{t|t-1} H^{(i)}_t (H^{(i)}_t P^{(i)}_{t|t-1} H^{(i)\prime}_t + R^{(i)}_t)^{-1} \qquad (3.13)$$

$$P^{(i)}_{t+1|t} = F_t P^{(i)}_{t|t-1} F'_t + Q_t - K^{(i)}_t (H^{(i)}_t P^{(i)}_{t|t-1} H^{(i)\prime}_t R^{(i)}_t) K^{(i)\prime}_t \qquad (3.14)$$

$$P^{(i)}_{0|-1} = \Pi_0 \quad .$$

There are many other ways to write the equations (3.12) to (3.14), which is optimal is not the issue in this paper, since only local processing gets involved. The local estimates of (3.12) are now combined to form the estimate given all the data $Z_t = \bigcup_{i=1}^{r} Z^{(i)}_t$ as:

$$\hat{x}_{t+1|t} = F_t \sum_{i=1}^{r} P_{t|t} \left[((P^{(i)}_{t|t-1})^{-1} + H^{(i)\prime}_t (R^{(i)}_t)^{-1} H^{(i)}_t) F^{-1}_t \hat{x}^{(i)}_{t+1|t} + h^{(i)}_t \right] \qquad (3.15)$$

where $P_{t|t}$ is the global filtered estimate error covariance

$$P^{-1}_{t|t} = P^{-1}_{t|t-1} + \sum_{i=1}^{r} (H^{(i)}_t)' (R^{(i)}_t)^{-1} H^{(i)}_t \qquad (3.16)$$

$$P_{t+1|t} = F_t P_{t|t} F'_t + Q_t \quad ; \quad P_{0|-1} = 0 \qquad (3.17)$$

and $h^{(i)}_t$ is the additional data-dependent n-vector which is locally updated from:

$$h^{(i)}_{t+1} = \Lambda_{t+1} h^{(i)}_t + \Gamma^{(i)}_{t+1} \hat{x}^{(i)}_{t+1|t} \quad ; \quad h^{(i)}_0 = 0 \qquad (3.18)$$

where:

$$\Lambda_t = P_{t|t} P^{-1}_{t|t-1} F_t \tag{3.19}$$

$$\Gamma^{(i)}_t = P_{t|t}\left[P^{-1}_{t|t-1} F_t P_{t-1|t-1} (P^{(i)}_{t-1|t-1})^{-1} F^{-1}_{t-1} - (P^{(i)}_{t|t-1})^{-1}\right] . \tag{3.20}$$

In the original paper F_t was taken to be constant. The method fails when F_t is singular at some time. As indicated earlier, only data acquired at node N_i is needed for the calculation of $\hat{x}^{(i)}_{t+1|t}$ and $h^{(i)}_{t+1}$, but the computation of $h^{(i)}_{t+1}$ assumes the knowledge of $P_{t|t}$ and $P_{t|t-1}$ a priori, through the formulas (3.19) and (3.20). From equation (3.16) this boils down to the a priori knowledge of the observation process of <u>all</u> nodes N_i!

Another disadvantage of this method is the fact that by breakdown the computed best <u>global</u> estimate is no longer (sub)optimal, although locally suboptimal estimates are computed. Indeed, if as suggested by Speyer the $P_{t|t}$ and/or $P_{t|t-1}$ are precomputed and then fixed, the global estimate (3.15) will only be optimal if all terms in the summation are accounted for (i.e., if all nodes communicated their $\hat{x}^{(i)}_{t+1|t}$ and $h^{(i)}_t$). A modification which would still yield suboptimal global estimates will have to compute the $P_{t|t}$ in <u>real time</u> and based on the effectively linked stations in the network at the given time. This will again require transmission of this real time computed global error-covariance to all nodes, so that robust estimates can be formed. We can conclude that the above-discussed algorithm is efficient in the context of distributed control, where it is the goal to obtain fault-tolerant local controllers, which were suboptimal when there was a breakdown in one or more of the links of the system. The algorithm is communication cost-effective if the observation parameters are known a priori. The global estimate is not robust.

3.3 Decentralized Estimation with Unknown Observation Parameters

If we pursue a different goal: an optimal global estimate which remains suboptimal when data from one or several nodes is interrupted, and which requires the least amount of data transmission, the following algorithm can be used. The development is along the same lines as in Section 2, although the results are believed to be new.

To account for optimality in the presence of failures, we will assume that the number of participating nodes is time variant. Thus in (3.6) we set $r = r(t)$. Moreover, we will assume that only at time t, the value of $r(t)$ is known. In this situation, the global error-covariance P cannot be precomputed as is needed in Speyer's method. We will give the solution for the time update and the measurement update separately, basically because for the latter, we can use the results obtained in Section 2.

Measurement Update

At stage t, we suppose that $r(t)$ stations perform the measurements (3.6). Previous to this the best estimate (globally) of the state vector $x(t)$ is $\hat{x}(t|t-1)$ (i.e., the optimal estimate given data up to t-1) with error-covariance $P_{t|t-1}$.

First, we reduce this problem to the parameter estimation problem of Section 2. Introducing "normalized" observations:

$$\eta_t^{(i)} \triangleq [R_t^{(i)}]^{-1/2}(y_t^{(i)}-H_t^{(i)}\hat{x}(t|t-1)) \qquad (3.21)$$

we obtain:

$$\eta_t^{(i)} = [R_t^{(i)}]^{-1/2} H_t^{(i)} (x(t)-\hat{x}(t|t-1))+[R_t^{(i)}]^{-1/2} v_t^{(i)} \qquad (3.22)$$

and it is easy to check that the a priori $\tilde{x}(t|t-1) \overset{\Delta}{=} x(t)-\hat{x}(t|t-1)$ has zero mean and covariance $P_{t|t-1}$ while the second term in (3.22) satisfies the conditions (2.5) and (2.6). Defining $\bar{H}_t^{(i)}$ and $\bar{v}_t^{(i)}$ as in (2.4), the equivalent measurement is:

$$\eta_t^{(i)} = \bar{H}_t^{(i)} \tilde{x}(t|t-1) + \bar{v}_t^{(i)} \qquad . \qquad (3.23)$$

From Section 2, the measurement update -- incorporating the measurements at time t -- yields by definition the estimates $\hat{x}(t|t)$, and is given by equation (2.12)

$$\tilde{x}(t|t) = (P_{t|t-1}^{-1} + \sum_{i=1}^{r(t)} \bar{H}_t^{(i)\prime} \bar{H}_t^{(i)})^{-1} \sum_{i=1}^{r(t)} \bar{H}_t^{(i)\prime} \eta_t^{(i)} \qquad (3.24)$$

or, after back-substitution to the original parameters:

$$\hat{x}(t|t) = (P_{t|t-1}^{-1} + \sum_{i=1}^{r(t)} H_t^{(i)\prime} (R_t^{(i)})^{-1} H_t^{(i)})^{-1} (\sum_{i=1}^{r(t)} H_t^{(i)\prime} (R_t^{(i)})^{-1} y_t^{(i)} + P_{t|t-1}^{-1} \hat{x}(t|t-1)) \qquad (3.25)$$

If, based on the (global) estimate $\hat{x}(t|t-1)$ and the data $y_t^{(i)}$, we want a "local" estimate at node (i), then we have:

$$\hat{x}^{(i)}(t|t) = (P^{-1}_{t|t-1}+H^{(i)\prime}_t(R^{(i)}_t)^{-1}H^{(i)}_t)^{-1}(H^{(i)\prime}_t(R^{(i)}_t)^{-1}y^{(i)}_t + P^{-1}_{t|t-1}\hat{x}(t|t-1)) \qquad (3.25)$$

The error-covariances of the global and the local estimate are respectively:

$$P_{t|t} = \left[P^{-1}_{t|t-1} + \sum_{i=1}^{r(t)} H^{(i)\prime}_t (R^{(i)}_t)^{-1} H^{(i)}_t \right]^{-1} \qquad (3.27)$$

$$P^{(i)}_{t|t} = \left[P^{-1}_{t|t-1} + H^{(i)\prime}_t (R^{(i)}_t)^{-1} H^{(i)}_t \right]^{-1} \qquad (3.28)$$

Remark 1: We assumed that the global estimate $P_{t|t-1}$ is always available since this was our prime goal. Local estimates are not necessarily computed.

Remark 2: From (3.27) and (3.28) the following relation between local and global estimates can be proven:

$$P^{-1}_{t|t} = \sum_{i=1}^{r(t)} [P^{(i)}_{t|t}]^{-1} + (1-r(t))P^{-1}_{t|t-1} \quad . \qquad (3.29)$$

It is now clear what preprocessing can be done locally in order to compress the data. At each node, the data vector $y^{(i)}_t$ and parameters $H^{(i)}_t$ and $R^{(i)}_t$ are combined to retain the "compressed" data:

$$z^{(i)}_t \overset{\Delta}{=} H^{(i)\prime}_t (R^{(i)}_t)^{-1} y^{(i)}_t \qquad (3.30)$$

and

$$L_t^{(i)} \overset{\Delta}{=} H_t^{(i)\prime}(R_t^{(i)})^{-1}H_t^{(i)} \quad . \tag{3.31}$$

Several different situations may occur, whether or not the communication between the central station and the link N_i is two-way, one-way or broken down.

<u>CASE (1)</u>: Station N_i is linked to the network and global information is accessible.

According to (3.28), $P_{t|t}^{(i)}$ can be computed at N_i, and a local estimate (3.26) may be computed "for later reference" (i.e., if at the next step the station should be disconnected from the net). To the central station, or one of the nodes performing as such, $z_t^{(i)}$ and $P_{t|t}^{(i)}$ or $L_t^{(i)}$ is sent. The latter matrices are symmetric of size n x n. Therefore they contain only $\frac{n(n+1)}{2}$ independent parameters. In total $\frac{n(n+1)}{2} + np_i$ parameters have to be transmitted.

<u>CASE (2)</u>: Station N_i is linked to the network, but has no information back from it.

In this case the local error covariance cannot be computed. Thus $z_t^{(i)}$ and $L_t^{(i)}$ have to be sent. A suboptimal local estimate can still be calculated. In many cases some a priori information about x_t will be available (e.g., from a previous time update). If absolutely nothing is known, it means that the a priori estimate can be arbitrary, but with infinite variance. This is equivalent to setting $P_{t|t-1}^{-1}$ equal to zero in the formulas (3.26) and (3.28).

The local estimate is:

$$\hat{x}^{(i)}(t|t) = (\bar{P}^{(i)}_{t|t})^{-1}H^{(i)\prime}_t(R^{(i)}_t)^{-1}y^{(i)}_t \qquad (3.32)$$

where we set:

$$\bar{P}^{(i)}_{t|t} = H^{(i)\prime}_t(R^{(i)}_t)^{-1}H^{(i)}_t = L^{(i)}_t \qquad (3.33)$$

Thus $L^{(i)}_t$ is to be interpreted as the local error-covariance in the absence of a priori information.

CASE (3): Station N_i does not contribute to the network.

Local estimates can be updated (suboptimal) with or without the access to $\hat{x}(t|t-1)$ and $P_{t|t-1}$. At the central node, global updating only accounts for the cooperating nodes, i.e., the cases (1) and (2) above. If nodes N_i; i=1,...,r(t) are these cooperating nodes at t, then equations (3.25) and (3.27) (or (3.29)) are used to compute a global filtered estimate.

Time Update

The next step is to compute the new predicted estimates $\hat{x}(t+1|t)$ from the filtered estimates $\hat{x}(t|t)$. Given all data up to time t, we write from (3.1)

$$\hat{x}_{t+1|t} = F_t\hat{x}_{t|t} + \hat{u}_{t|t} \quad . \tag{3.34}$$

Since u_t is uncorrelated with the previous noise u_s and v_s for $s < t$, and with $\bar{x}_0$, we have that $\hat{u}_{t|t} = 0$. The time update is therefore simply:

$$\hat{x}_{t+1|t} = F_t\hat{x}_{t|t} \tag{3.35}$$

and

$$P_{t+1|t} = F_tP_{t|t}F'_t + Q_t \tag{3.36}$$

for the global estimates, while the local time-updates are:

$$\hat{x}^{(i)}_{t+1|t} = F_t\hat{x}^{(i)}_{t|t} \tag{3.37}$$

$$P^{(i)}_{t+1|t} = F_tP^{(i)}_{t|t}F'_t + Q_t \quad . \tag{3.38}$$

If the emphasis is on an optimal global estimate, the centralized node updates from (3.34) and (3.35). A reason for local computation of estimates is for robustness and optimality. If for instance N_1 is cut-off from the network but local estimates are updated, then the global estimate can be upgraded with this "combined" information at N_1 as soon as the link is restored, and thus not all the data is lost.

The above discussion indicates that there are many ways of organizing the data traffic, and it seems fruitless, if not impossible to give all details.

4. THE DYNAMIC ESTIMATION PROBLEM: BLOCK PROCESSING

4.1 Problem Statement

For the problem, described in Section 3.1, we give here an optimal solution in case one does not need the estimates at each time instant, or in case the communication is impossible during intervals of fixed length. In the previous section we mentioned the case of an "isolated" node. During its cut-off time, the new incoming data was gathered and processed to update the old local data. These resulting data were transmitted at the time the link was restored, and incorporated in an optimal "forward" way. At that point the globa- estimate will not equal the optimal estimate provided by the Kalman filter that had access to all data y_t at time t! This loss of accuracy is balanced by a gain in communication and time. Indeed to provide the best estimate given all the data, including the batch of missed data, the global estimator would have to recompute previous error covariances and estimates, thus momentarily slip back to the instant the node was cut-off and reprocess the accumulated data step-by-step.

We present here an alternative method for the case where the system parameters are a priori known, and each node has a copy of this available. The data is gathered in batches at each node, and synchronized with all stations at the end of the interval, the accumulated processed data is sent to the global estimator, which then proceeds to compute the optimal global estimate. For the development of the algorithm, our scattering approach to the filtering and smoothing problem is used [7] since it provides a direct insight into the method.

4.2 A Distributed Block-Processing Algorithm

We have seen that the quantities $H_s' y_s = \sum_{i=1}^{r} H_s^{(i)'} y_s^{(i)} \overset{\Delta}{=} \sum_{i=1}^{r} z_s^{(i)}$ and $H_s' H_s = \sum_{i=1}^{r} H_s^{(i)'} H_s^{(i)} \overset{\Delta}{=} \sum_{i=1}^{r} L_s^{(i)}$ lead to a distributed computation in a natural way.

In a step by step updating algorithm, then, we compute locally the $L_s^{(i)}$ and $z_s^{(i)}$, transmit these data to the central station, where now the updating occurs (i.e., one 'layer' is added to the scattering picture.)

Further distributing is possible if we can divide the whole estimation interval (τ,t) up in some subintervals of each, say, σ steps long. The σ elementary scattering layers can thus be combined to a scattering layer representing such a subinterval. The 'parameters' are:

$$S(s+\sigma,s) = \begin{bmatrix} \Phi(s+\sigma,s) & P(s+\sigma,s) \\ -\mathcal{O}(s+\sigma,s) & \Psi(s+\sigma,s) \end{bmatrix} \qquad (4.1)$$

These parameter-matrices can be computed if (F_s,G_s,H_s) is known. Further, we can also combine all sources into a $q_o^-(s+\sigma,s)$ and $q_o^+(s+\sigma,s)$.

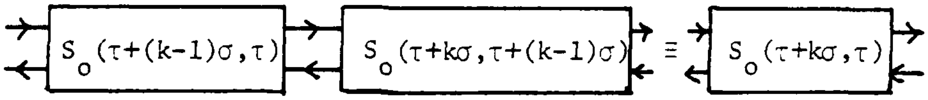
Suppose now that our model is stationary. Then for each 'time slot' the Φ, P, Θ and $\psi = \Phi'$ are the same, and so are the gains G_i^+ and G_i^- $i = 1, \ldots, \sigma$ where we define

$$\begin{aligned} q_o^+(s+\sigma, s) &= \sum_{i=1}^{\sigma-1} G_i^+ \cdot z(s+i) \\ q_o^-(s+\sigma, s) &= \sum_{i=1}^{\sigma-1} G_i^- \cdot z(s+i) \end{aligned} \tag{4.2}$$

Now, we let each station compute <u>locally</u>: $q_o^{+(i)}$ and $q_o^{-(i)}$ in a certain subinterval (say k) (which is done sequentially). At the end of the subinterval (of length σ) the "data" $q_o^{-(i)}$ and $q_o^{+(i)}$ from all stations N_i are transmitted to a 'central location', where the data is combined and the whole scattering layer is formed, i.e.,

$$q_o^- = \sum_{i=1}^{r} q_o^{-(i)} \quad , \qquad q_o^+ = \sum_{i=1}^{r} q_o^{+(i)} \tag{4.3}$$

While the stations proceed now gathering more data and computing the values for q_o^- and q_o^+ in the subsequent subinterval, the central station combines the (combined) picture of the previous k-1 subintervals with the k^{th} one, thus obtaining the $\phi, \psi, P, \Theta, q_o^-$ and q_o^+ over the interval $(\tau, \tau+k\sigma)$ (Fig. 4.1).


$S_o(\tau+(k-1)\sigma, \tau)$
$S_o(\tau+k\sigma, \tau+(k-1)\sigma)$
$\equiv$
$S_o(\tau+k\sigma, \tau)$

Finally, when the last subinterval is processed, the 'boundary layer' representing the initial condition is attached, and the estimates are computed. We can represent the above procedure on a space/time flowchart, the vertical direction being time, the horizontal direction the space. For clarity we only display one substation, (Fig. 4.2).

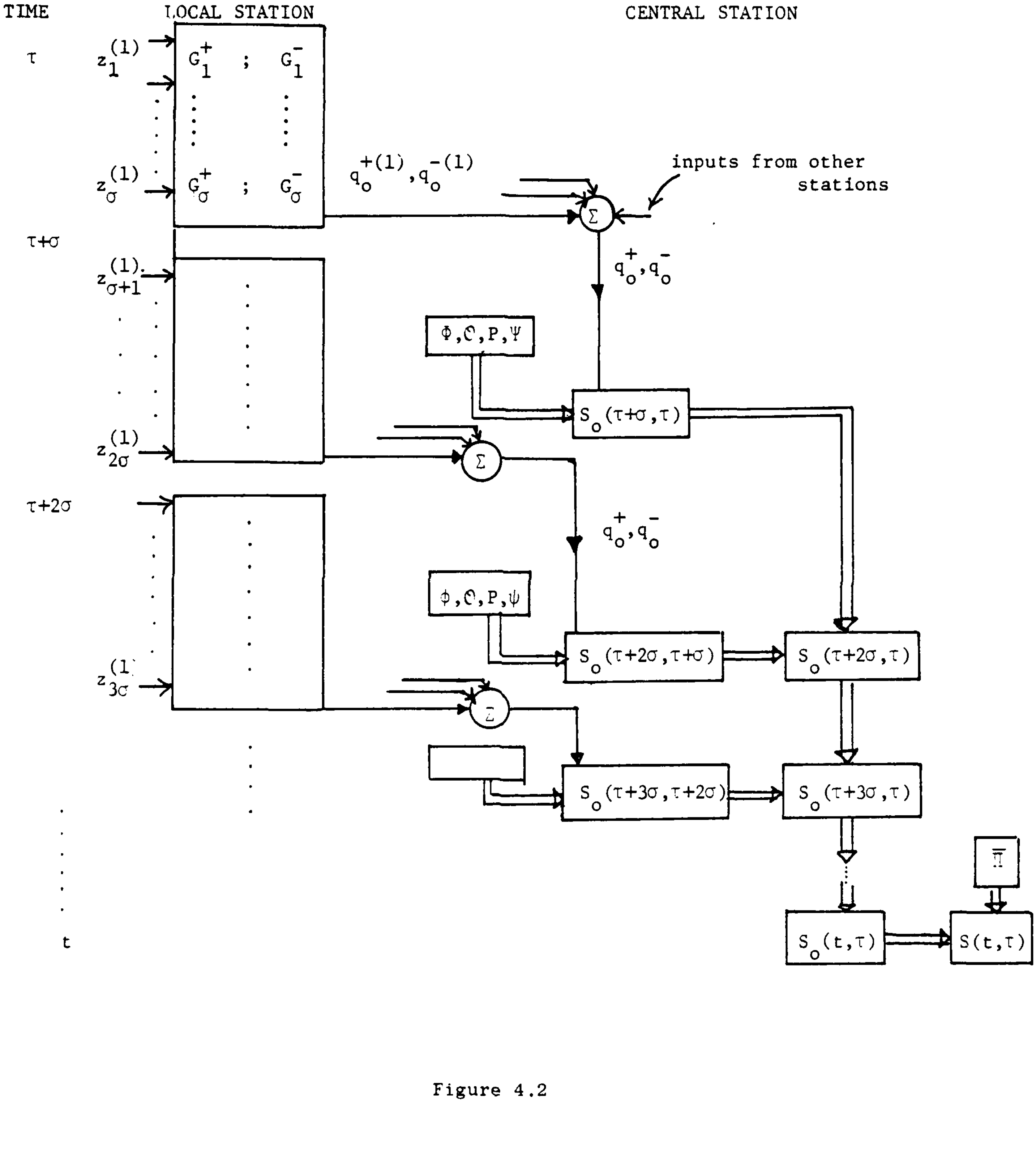

TIME
LOCAL STATION
CENTRAL STATION
τ
$z_1^{(1)}$
G_1^+ ; G_1^-
$z_\sigma^{(1)}$
G_σ^+ ; G_σ^-
$q_o^{+(1)}, q_o^{-(1)}$
inputs from other stations
Σ
τ+σ
q_o^+, q_o^-
$z_{\sigma+1}^{(1)}$
Φ,Θ,P,Ψ
$S_o(\tau+\sigma,\tau)$
$z_{2\sigma}^{(1)}$
Σ
τ+2σ
q_o^+, q_o^-
ϕ,Θ,P,ψ
$S_o(\tau+2\sigma,\tau+\sigma)$
$S_o(\tau+2\sigma,\tau)$
$z_{3\sigma}^{(1)}$
Σ
$S_o(\tau+3\sigma,\tau+2\sigma)$
$S_o(\tau+3\sigma,\tau)$
Π̄
t
$S_o(t,\tau)$
$S(t,\tau)$

Figure 4.2

The Algorithm:

1) Precomputation of the "Data-Gains" and "Block Parameters"

If the parameters are not constant, they have to be recombined in each interval by iteration of the scattering formulas in order to obtain the "block-scattering parameters" subject to the initial conditions [7]. In Fig. 4.2 this is indicated by the box containing $(\Phi, \mathcal{O}, P, \Psi)$. The gains G_s^+; G_s^- for $s = 1,\ldots,\sigma$ are also obtained by iteration [7]. All nodes are supposed to have these parameters available.

2) Local Data Processing

At the beginning of each block we reset $q_s^{+(i)}$ and $q_s^{-(i)}$ to zero. As data comes in, we proceed step-by-step

$$z_s^{(i)} = H_s'^{(i)} y_s^{(i)}$$

$$q_s^{\pm(i)} = q_s^{\pm(i)} + G_s^{\pm} z_s^{(i)}$$

until $s = \sigma$ when the contents of $q_s^{+(i)}$ and $q_s^{-(i)}$ registers are sent to the central station.

3) Central Computation (kth Block)

The new block is added to the previously combined k-1 blocks. As shown in Fig. 4.1, the composite is given by the Redheffer "star-product" [10]:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A(I-bC)^{-1}a & B+Ab(I-Cb)^{-1}D \\ c+dC(I-bC)^{-1}a & d(I-Cb)^{-1}D \end{bmatrix} \qquad (4.4)$$

The internal "sources" are similarly combined:

$$\begin{bmatrix} Q^+ \\ Q^- \end{bmatrix} \triangleq \begin{bmatrix} q^+ \\ q^- \end{bmatrix} + \begin{bmatrix} r^+ \\ r^- \end{bmatrix} \triangleq \begin{bmatrix} r^+ \\ q^- \end{bmatrix} + \begin{bmatrix} A(I-bC)^{-1}(q^+ + br^-) \\ d(I-Cb)^{-1}(r^- + Cq^+) \end{bmatrix} \qquad (4.5)$$

Finally, the filtered estimates $\hat{x}(t|t)$ at the end of the k^{th} block are found by "appending" the end-conditions $\lambda(t|t) = 0$. The initial ("boundary layer" accounts for the a priori knowledge. This layer is represented by:

$$\bar{\Pi} = \begin{bmatrix} I & \Pi_0 \\ 0 & I \end{bmatrix} \qquad (4.6)$$

Comparison of number of data transmissions in Section 4 and Section 3.

1. Step by step Kalman Filter.

   Each sampling instant requires $\frac{n(n+1)}{2} + n$ numbers to be sent.

   Thus, in total for an interval of length $t - \tau$

   $$\# = (t-\tau) \cdot \frac{n^2 + 3n}{2}$$

2. 'Partitioned' filter.

   (Note that we discussed the smoothing problem here, but the filtering formulas, and therefore the procedure, are identical in form).

   There are now $\frac{t - \tau}{\sigma}$ transmissions required of the 'data' q_o^+ and q_o^- 2 vectors of size n. Thus:

   $$\# = \frac{t - \tau}{\sigma} \cdot 2n$$

5. DISTRIBUTED DETECTION ALGORITHMS

A common problem in sensor-systems is to decide whether a signal is picked up, imbedded in noise, or if we only "observe" the noise. These signal-detection problems arise in radar, sonar and seismic sensors. There are two hypotheses:

$\mathcal{H}_0$: signal plus noise is observed;

$\mathcal{H}_1$: noise only is observed.

More general we shall consider the case where there are M hypotheses. The observations are then likewise used to decide which of the M models corresponds "best" to these data. There are many other applications for hypothesis-testing; for instance: component failure detection, pattern recognition, system identification [9]. Details on the theory can be found in many books, e.g., [9,], [11], and [12]. Here we shall look at some particular cases.

5.1 Static Systems Hypothesis Testing

Suppose again that we have r disjoint sensors N_i; $i=1,\ldots,r$ where observations $y^{(i)}$ are made. Let us have the following M hypotheses (at each station):

$$\mathcal{H}_j^{(i)}: \quad y^{(i)} = H_j^{(i)} x_j + v_j^{(i)} \quad , \quad j=1,\ldots,M \tag{5.1}$$

where

$$x_j \text{ is } N(0,\Pi_j) \tag{5.2}$$

$$v_j^{(i)} \text{ is } N(0,R_j^{(i)}) \tag{5.3}$$

and further x_j and $v_j^{(i)}$ independent, and $v_j^{(i)}$-independent from all $v_k^{(i)}$ if $\ell \neq i$.

Clearly, the hypotheses are about the same (global) parameter x that may assume the values $x_1 \ldots x_M$. Therefore if we consider the combined problem by stacking together the observations from each node, then we have not Mr but still r hypotheses.

$$\mathcal{H}_j: \quad y = H_j x_j + v_j \quad , \quad j=1,\ldots,M \tag{5.4}$$

where y_j, H_j and v_j are defined as in (2.7) and

$$E\, v_j v_j' = \text{Block diagonal } [R_j^{(1)} \ldots R_j^{(r)}] \tag{5.5}$$

It is intuitively clear that by combining data from r sensors, better decisions will be made than if each detector would "decide" based on its own data. The log likelihood function ξ_j for the j^{th} hypothesis is [9]:

$$2\xi(y) = c_j - y'(H_j \Pi_j H_j' + R_j)^{-1} y \quad , \tag{5.6}$$

where c_j is a data-independent constant determined by such factors as a priori information on the probability of the hypothesis and the penalties for incorrect decisions. In general, the decision is then to take the hypothesis $\mathcal{H}_k$ if $j=k$ yields maximum $\xi_j(z)$. For a binary decision ($M=2$) the log likelihood ratio ℓ is more useful:

$$\ell(y) = \xi_1(y) - \xi_2(y) \quad . \tag{5.7}$$

The equivalent decision rule is then to compare $\ell(y)$ to a threshold λ which depends again on the a priori probabilities and the costs of incorrect decisions.

$$\text{Choose } \mathcal{H}_1 \quad \text{if} \quad \ell(y) \geq \lambda$$

$$\text{Choose } \mathcal{H}_2 \quad \text{if} \quad \ell(y) < \lambda \quad .$$

The similarity in the forms of equations (5.6) and (2.10) illustrate the close relation between detection and estimation, and accordingly distributed schemes exist.

Equation (5.6) can indeed be rewritten as:

$$2\xi_j(y) = c_j - y'R_j^{-1}y - y'R_j^{-1}H_j(\Pi^{-1} + H_j'R_j^{-1}H)^{-1}H_jR_j^{-1}y \tag{5.8}$$

This form makes clear how the computation may be distributed using methods very similar to what we used in the estimation problem. The local computation at node N_i involves:

$$\left.\begin{aligned} L_j^{(i)} &= H_j^{(i)\prime}(R_j^{(i)})^{-1}H_j^{(i)} \\ \mu_j^{(i)} &= y^{(i)\prime}(R_j^{(i)})^{-1}y^{(i)} \\ z_j^{(i)} &= H_j^{(i)\prime}(R_j^{(i)})^{-1}y^{(i)} \end{aligned}\right\} \quad j=1,\dots,M \qquad (5.9)$$

The central station then assembles the log likelihood functions:

$$2\xi_j(y) = c_j - \sum_{i=1}^{r} \mu_j^{(i)} - \Big(\sum_{i=1}^{r} z_j^{(i)}\Big)'\Big(\Pi^{-1} + \sum_{i=1}^{r} L_j^{(i)}\Big)^{-1}\Big(\sum_{i=1}^{r} z_j^{(i)}\Big) \qquad (5.10)$$

and compares them against each other, to decide which hypothesis should be taken.

Transmission of the $L_j^{(i)}$, $\mu_j^{(i)}$ and $z_j^{(i)}$; $i=1,\dots,r$; $j=1,\dots,M$ involves maximally $M\left(\frac{n(n+1)}{2} + 1 + n\right)$ parameters compared to the $\sum_{i=1}^{r} p_i + M \sum_{i=1}^{r}\left(p_i n + \frac{p_i(p_i+1)}{2}\right)$ that are required for efficient transmission of the $y^{(i)}$, $H_j^{(i)}$ and $R_j^{(i)}$. From the point of view of data transmission, one prefers distribution if:

$$n^2 + 3n + 2 \le (2n + 1 + 2/M) \sum_{i=1}^{r} p_i + \sum_{i=1}^{r} p_i^2 \quad . \qquad (5.11)$$

5.2 Dynamic Systems: Hypothesis Testing

Just as in the static case, we may expect a close similarity between estimation and detection and thus the aspects of distributed processing can also be fruitful in this case.

Of particular interest are the applications to system identification. Suppose we have M hypothesized linear system models:

$$x(t+1) = F_j(t)x(t) + u_j(t) \quad ; \quad j=1,\ldots,M \tag{5.12}$$

$$y^{(i)}(t) = H_j^{(i)}(t)x(t) + v_j^{(t)} \quad ; \quad i=1,\ldots,r \tag{5.13}$$

with: $u_j(t)$ = white noise with zero mean and covariance Q_j and $x(0)$ normally distributed with parameters $(0, \Pi_0)$. The assumptions on the observations (5.13) are as usual, and we may again combine them as in (5.4).

A decision at time t is essentially based (see [9, p. 286]) on computation of the M log likelihood functions, and on subsequent comparison.

The $\xi_j(t)$ are given by:

$$\xi_j(t) = -\sum_{s=1}^{t} (y(t)-H_j(t)\hat{x}_j(t|t-1))'(H_j(t)P_j(t|t-1)H_j'(t)+R_j(t))^{-1} \cdot (y(t)-H_j(t)\hat{x}_j(t|t-1)) \tag{5.14}$$

where $\hat{x}_j(t|t-1)$ is the predicted estimate under the hypothesis $\mathcal{H}_j$ and $P_j(t|t-1)$ is the corresponding error-covariance.

Various distributed algorithms for the estimation were already considered in previous sections, and (5.14) can again be rewritten involving the sums $\sum_{i=1}^{r} z_j^{(i)}(t)$, $\sum_{i=1}^{r} \mu_j^{(i)}(t)$ and $\sum_{i=1}^{r} L_j^{(i)}(t)$. We will omit the details. We end this secti on with the observation that in the case of non-a priori knowledge of H and R, the detection problem requires the additional scalars "μ" which were not needed in the corresponding estimation problem.

In the case of a priori knowledge (hypothesized) of $H_j^{(i)}$ and $R_j^{(i)}$, the data can be processed optimally in batches as in the estimation problem of Section 4.

6. CONCLUSIONS

Some algorithms for the distributed estimation and detection of both static and dynamic linear models with Gaussian modeled uncertainties were presented. The emphasis was on global estimates and decisions rather than on the local ones. The detection algorithms were shown to follow in a straightforward way from their estimation counterparts. Based on our scattering approach we derived an optimal distributed block processing algorithm for the case where each sensor has a copy of the system. These results are believed to be new. Further a new step-by-step procedure in the case of unknown observation parameters was developed. It allows for reconfigurability and survivability as the number of participating nodes may vary, while still suboptimal global estimates are given. If the global estimate and covariance are fed back to the nodes, virtually every node can perform the "centralized" computations, thus providing failure robustness. Finally, prprocessing at the sensor location may enclose some valuable information about the sensors or their location, a welcome feature in some applications.

REFERENCES

[1] E.J. Davison, N.K. Tripathi, "The Optimal Decentralized Control of a Large Power System: Lead and Frequency Control," IEEE Trans. on Autom. Contr., Vol. AC-23, No. 2, pp. 312-325, April 1978.

[2] "Distributed Sensor Nets," Proc. of the Dec. 7-8, 1978 Workshop, Carnegie-Mellon University.

[3] Special Issue on Large-Scale Systems and Decentralized Control, IEEE Trans. on Autom. Contr., Vol. AC-23, No. 2 , April 1978.

[4] J.L. Speyer, "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problems," CDC, Jan. 1979, pp. 1126-1131.

[5] D.D. Siljak, M.B. Vukcevic, "On Decentralized Estimation," Int. J. Contr., Vol. 27, No. 1, pp. 113-131, 1978.

[6] M.F. Hassan, "Optimum Kalman Filter for Large-Scale Systems Using the Partitioning Approach," IEEE Trans. SMC, Vol. SMC-6, No. 10, Oct. 1976.

[7] B. Friedlander, T. Kailath, L. Ljung, "Scattering Theory and Linear Least Squares Estimation - II: Discrete-Time Problems," J. Franklin Inst., Vol. 301, Nos. 1 & 2, Jan-Feb 1976.

[8] T. Kailath, "Lectures on Linear Least-Squares Estimation, Wien: Springer-Verlag, 1976.

[9] F.C. Schweppe, Uncertain Dynamic Systems, Englewood Cliffs, NJ, Prentice Hall, 1973.

[10] R. Redheffer, J. Math. Phys., XLI, pp. 1-41, 1962.

[11] H.L. Van Trees, Detection, Estimation and Modulation Theory, Part 1, John Wiley & Sons, 1968.

[12] A.P. Sage and J.L. Melsa, Estimation Theory with Applications to Communications and Control, McGraw-Hill, 1971.

[13] A.E. Bryson, Y.C. Ho, Applied Optimal Control, Ginn & Co., 1969.

APPENDIX A

A note on the transmission of the quantity $L^{(i)}$

We assume that x and $y^{(i)}$ are respectively n- and p_i-dimensional. Then $H^{(i)}$ has $p_i \cdot n$ parameters. $L^{(i)}$ is a symmetric $n \times n$ matrix, and therefore contains $\frac{n(n+1)}{2}$ independent parameters.

For this we could for instance take the lower triangular part of $L^{(i)}$, denoted as $[L^{(i)}]_+$. (Thus an additive decomposition of $L^{(i)}$), or we may consider a multiplicative decomposition of $L^{(i)}$ into its Cholesky triangular factors. We shall consider in greater detail why we might do this.

1. Additive Decompositions

Obviously to get $[L^{(i)}]_+$ from $L^{(i)}$ no additional computation at station N_i is required. If at station N_j the information of station N_i is to be added to the local information, then simply station N_j performs:

i) $L^{(j)} \rightarrow [L^{(j)}]_+$ local information

ii) $[\bar{L}^{(j)}]_+ = [L^{(j)}]_+ + [L^{(i)}]_+$ updating with information from N_i

Again step ii) is a simple set of $\frac{n(n+1)}{2}$ additions of scalars, and the result is in a direct transmittable form. The 'end station' needs to reconstruct the $\bar{L}$ from the $[\bar{L}]_+$: this is a simple information shift.

A note about secure transmission: For instance, in a tracking problem the parameters in $H^{(i)}$ depend on the local coordinates of the tracking station, and in some applications we may want to keep this secret. But now:

$$[L^{(i)}]_{+k\ell} = \sum_{m=1}^{p_i} H^{(i)}_{mk} \; H^{(i)}_{m\ell} \qquad k \geq \ell$$

This yields $\frac{n(n+1)}{2}$ equations for the $p_i n$ unknowns $H^{(i)}_{k\ell}$.

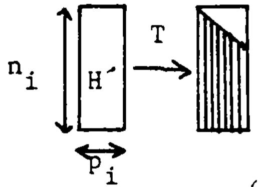
If $p_i \leq \frac{n+1}{2}$ these nonlinear equations can be solved, at least in principle, revealing the $H^{(i)}_{k\ell}$. Remember, however, that we opted for a distributed processing if $p_1 \geq \frac{n}{2} + 1$. To keep secrecy, we can inject redundancy (i.e., augmenting p_i). This has no additional effects on the transmission requirements, although it increases the (local) computational burden to compute $L^{(i)}$. Another possibility is to scramble the transmitted data $[L^{(i)}]_+$, but then it has to be descrambled.

2. Multiplicative Decomposition

Each station does a triangular factorization of the data $L^{(i)}$. Let $U^{(i)}$ be a lower triangular factor of $L^{(i)}$, then:

$$H'H = L = U\,U'$$

That means that at station N_i, we do not have to compute $L^{(i)}$ actually, but just a triangularization of H' (by column operations)



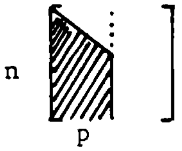
If station N_i transmits the data $U^{(i)}$ to N_j, then N_i can directly compute an update $U^{(j)}$ incorporating his own data $H^{(j)}$. Indeed, performing column operations on the array yields the lower triangular factor $U^{(j)}$

$$n_i \updownarrow [\underbrace{U^{(i)}}_{n_i} \vdots \underbrace{H^{(j)}}_{p_i}] \xrightarrow{T} [U^{(j)} \quad 0]$$

So, local information processing and updating (or combining) can all be done in one step. Further, if

$$H'H = U U'$$

then U is of the form



Let p be the number of nonzero columns, then p is the number of observations at the station, and these p columns are a scrambled version of H'. But again since $H'H$ is then known, we can solve for H if $p_i \leq \frac{n+1}{2}$.

Similarly, if we know the incoming and outgoing U to a station, then:

$$\xrightarrow{U^{(i)}} \boxed{N_j} \xrightarrow{U^{(j)}} \qquad U^{(j)}U^{(j)'} - U^{(i)}U^{(i)'} = H^{(j)'}H^{(j)}$$

and $H^{(j)}$ can be retrieved if the number of nonzero columns in the triangular factorization of $U^{(j)}U^{(j)'} - U^{(i)}U^{(i)'}$ is less or equal to $\frac{n+1}{2}$.

Thus sending the parameters in 'square-root' form does not provide us with higher secrecy nor simplicity, but might be preferred if the inverse of a triangular matrix is found by simple substitution, and the inverse of the square root of M is the square root of M^{-1}, whereas in general $[M]_+^{-1} \neq [M^{-1}]_+$.